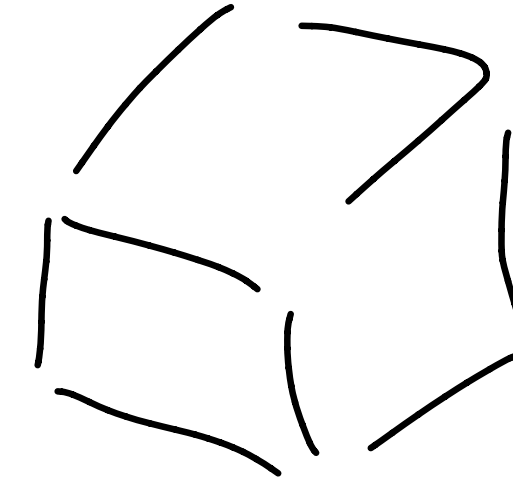


시뮬레이션

최백준 choi@startlink.io

주사위 굴리기

<https://www.acmicpc.net/problem/14499>



- 크기가 $N \times M$ 인 지도 위의 한 칸에 주사위가 놓여져 있다.
- 문제에 주어진 대로 주사위를 굴렸을 때, 상단에 써 있는 값을 매번 구하는 문제

주사위 굴리기

<https://www.acmicpc.net/problem/14499>

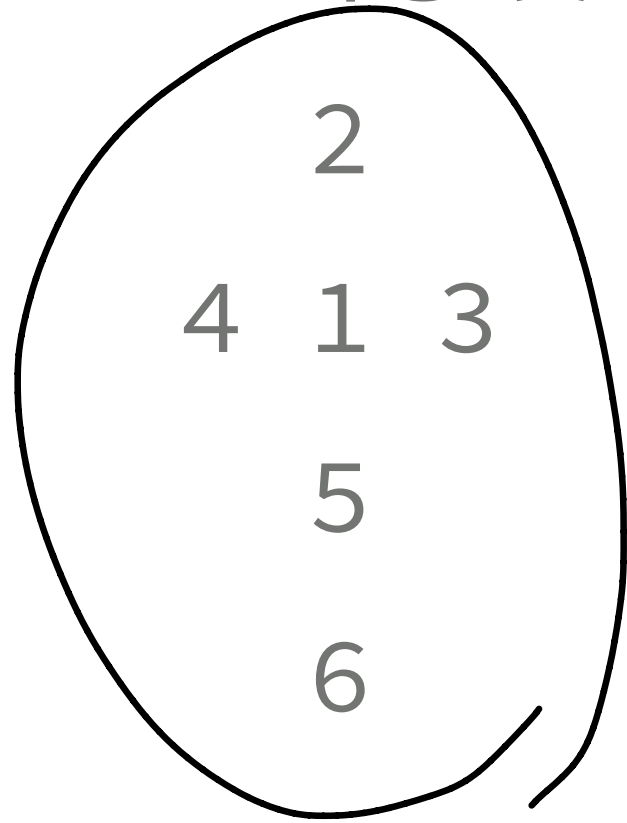
- 문제에 나와있는 대로 구현하는 문제는 조건을 모두 구현해야 한다.
- 구현해야 하는 조건
 - 가장 처음에 주사위에는 모든 면에 0이 적혀져 있음
 - 주사위를 굴렸을 때, 이동한 칸에 써 있는 수가 0이면, 주사위의 바닥면에 써 있는 수가 칸에 복사됨
 - 0이 아닌 경우에는 칸에 써 있는 수가 주사위의 바닥면으로 복사되며, 칸에 써 있는 수는 0이 ~~복사됨~~
 - 바깥으로 이동시키려고 하는 경우에는 해당 명령을 무시해야 함, 출력도 하면 안됨

5/2

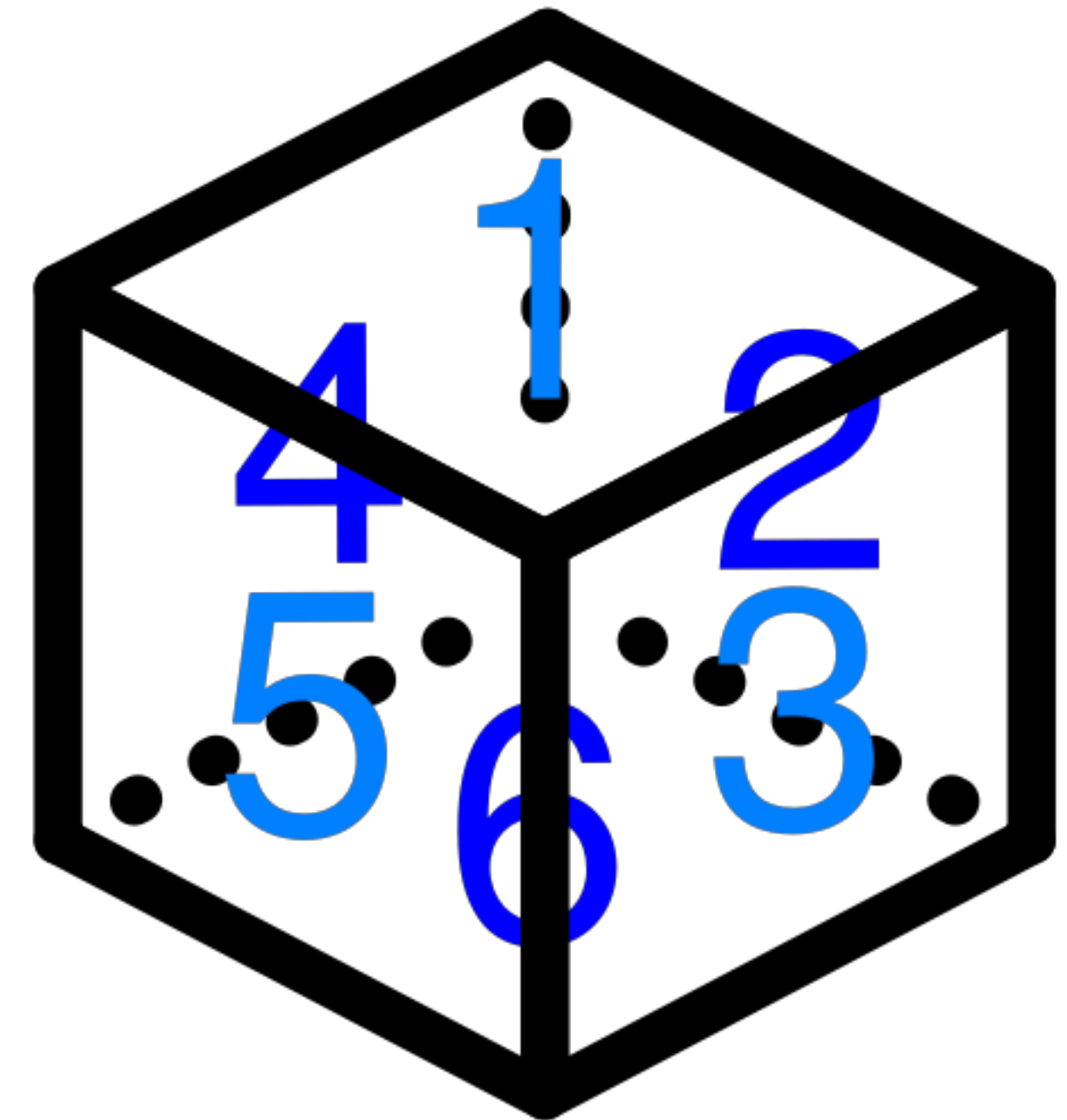
주사위 굴리기

<https://www.acmicpc.net/problem/14499>

- 가장 윗 면이 1, 동쪽을 바라보는 방향이 3이다.



- 각각의 이동에 대해서, 어떻게 면이 바뀌는지 알아봐야 한다.



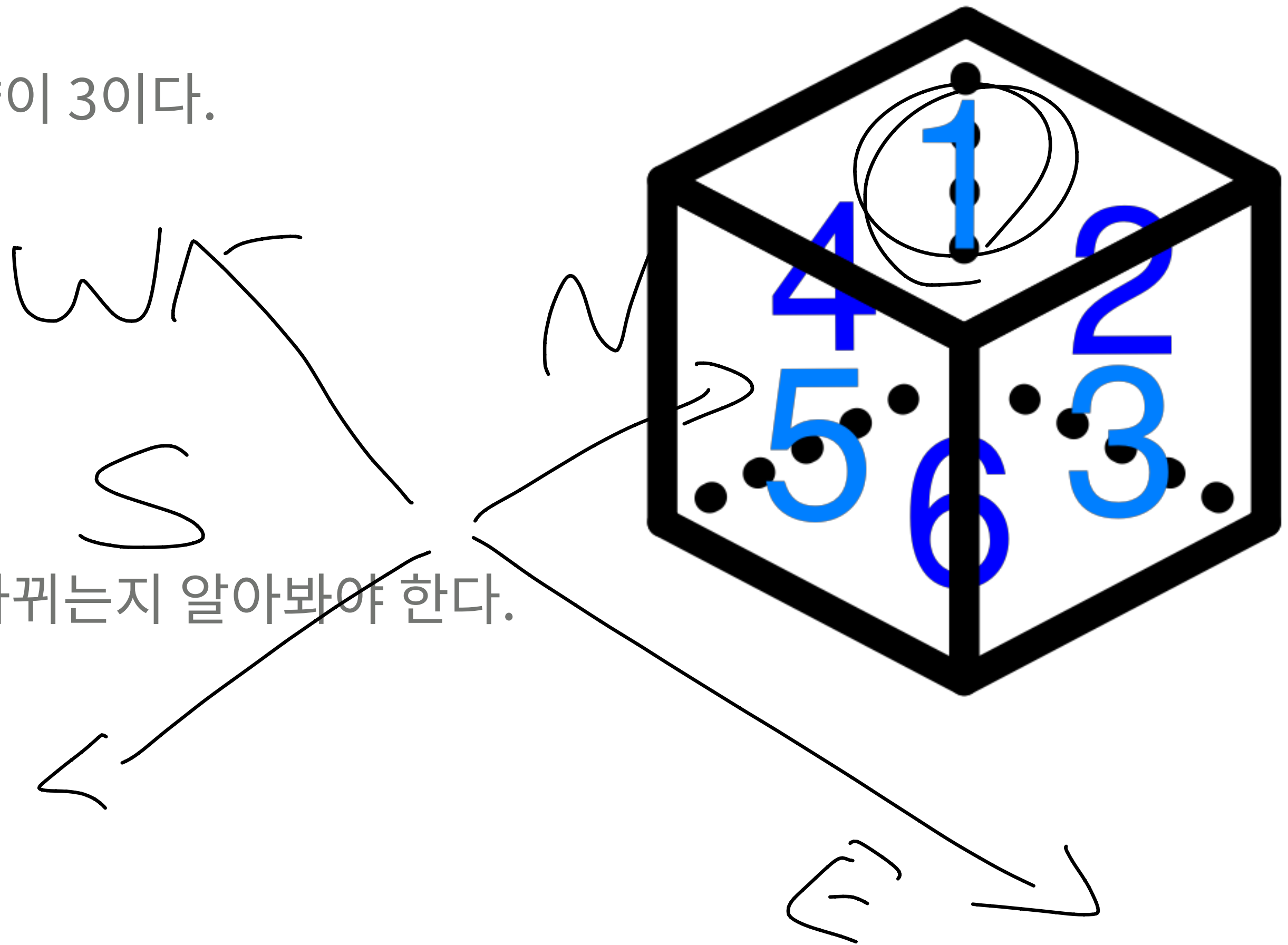
주사위 굴리기

<https://www.acmicpc.net/problem/14499>

- 가장 윗 면이 1, 동쪽을 바라보는 방향이 3이다.

2
4 1 3
5
6

- 각각의 이동에 대해서, 어떻게 면이 바뀌는지 알아봐야 한다.
- 동쪽 방향으로 굴리면 어떻게 될까?



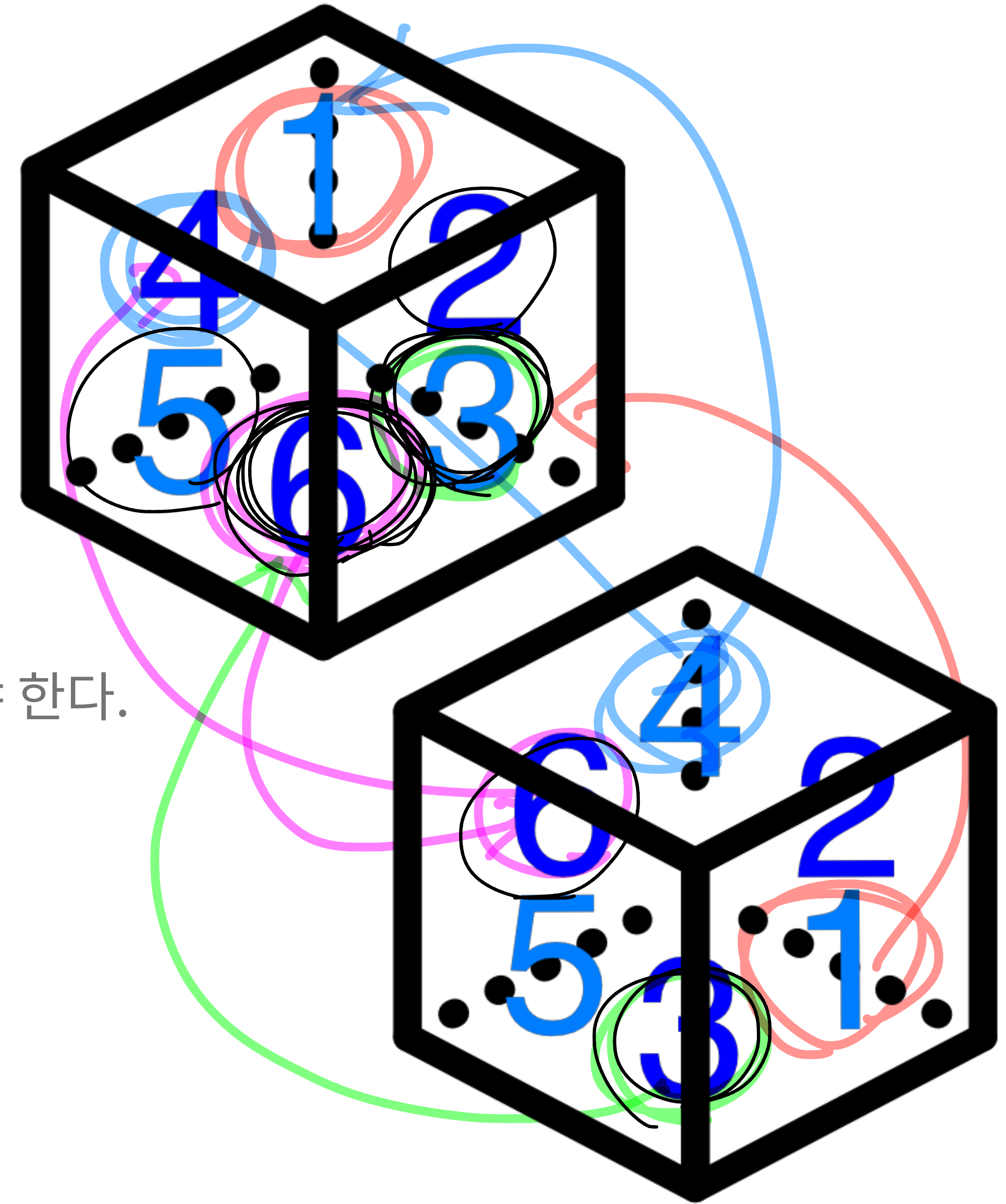
주사위 굴리기

<https://www.acmicpc.net/problem/14499>

- 가장 윗 면이 1, 동쪽을 바라보는 방향이 3이다.

2	→	2
4 1 3	→	6 4 1
5	→	5
6	→	3

- 각각의 이동에 대해서, 어떻게 면이 바뀌는지 알아봐야 한다.
- 동쪽 방향으로 굴리면 어떻게 될까?



주사위 굴리기

<https://www.acmicpc.net/problem/14499>

- 머리로 이걸 시뮬레이션 하는건 머리가 아프기 때문에
- 정육면체를 하나 만들어서 직접 돌려보는 것이 빠르다.

주사위 굴리기

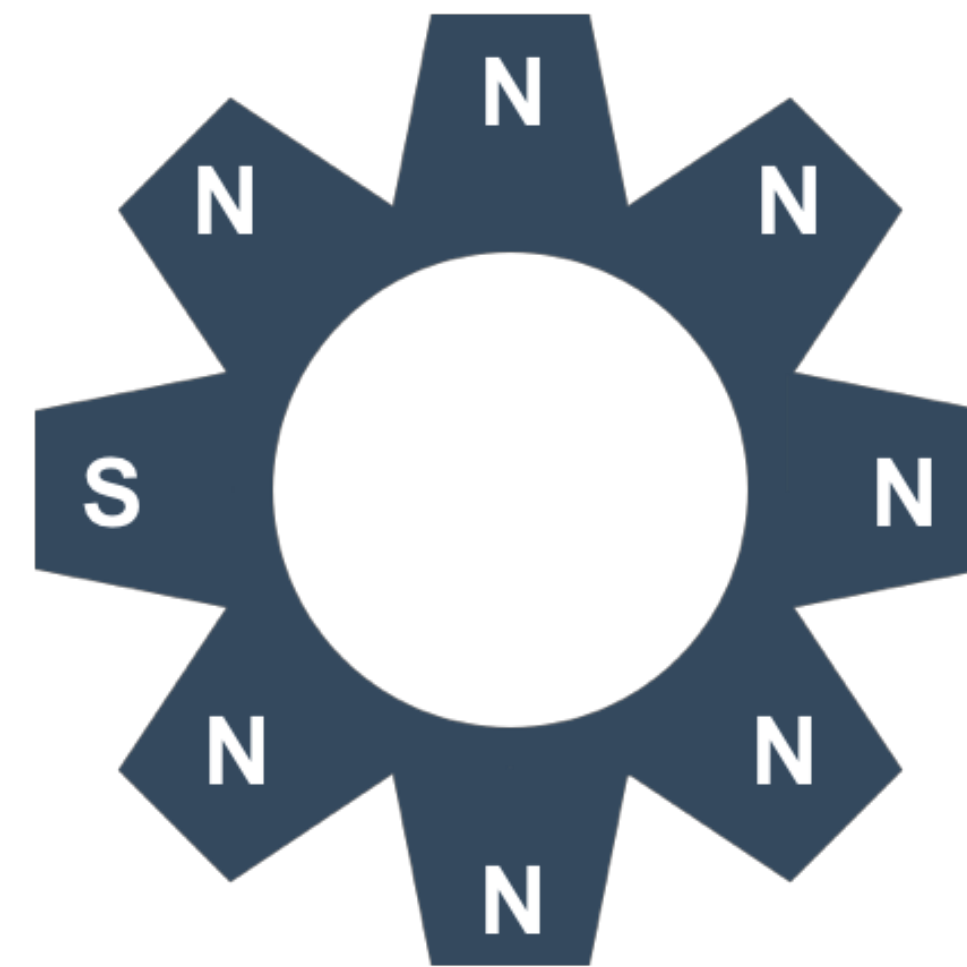
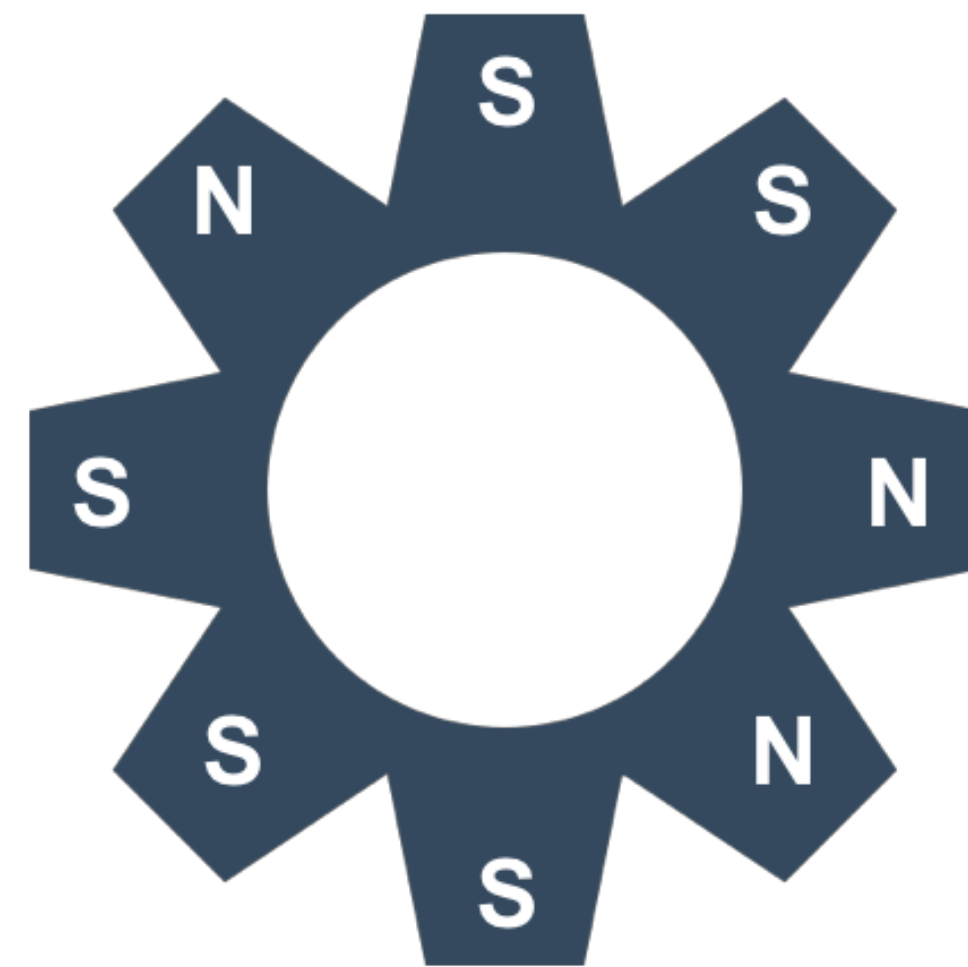
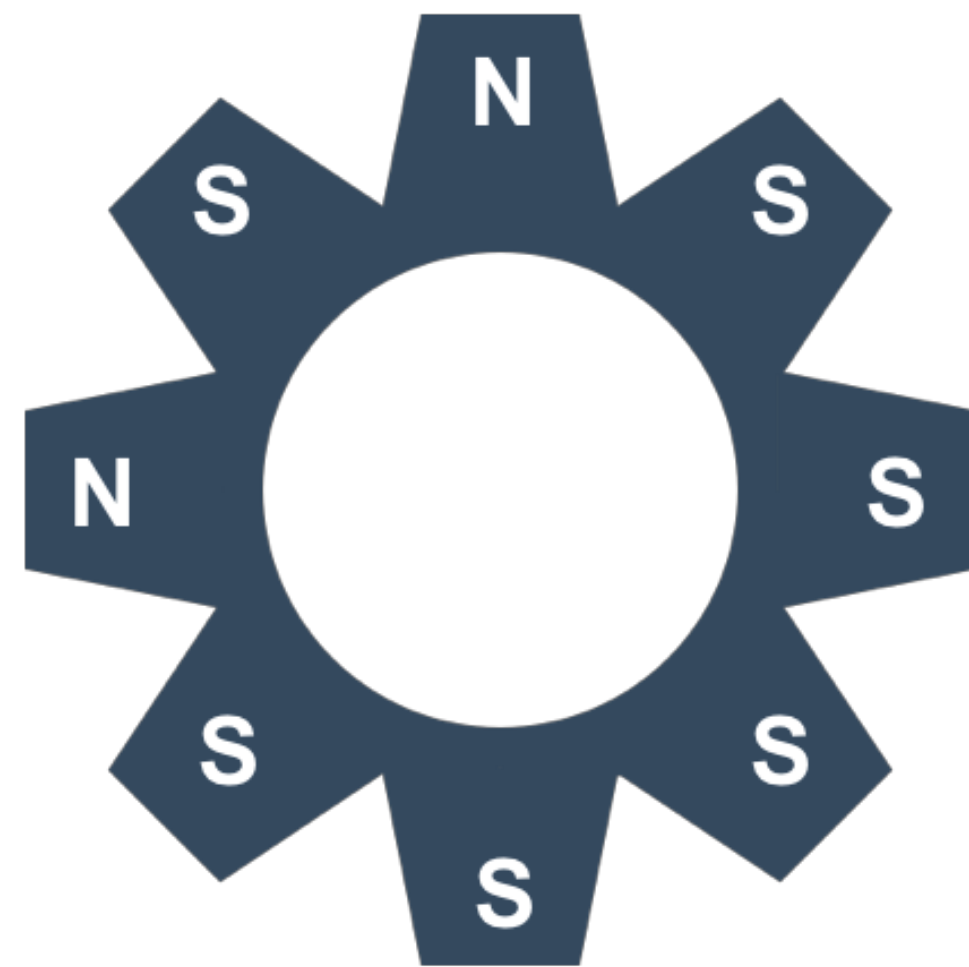
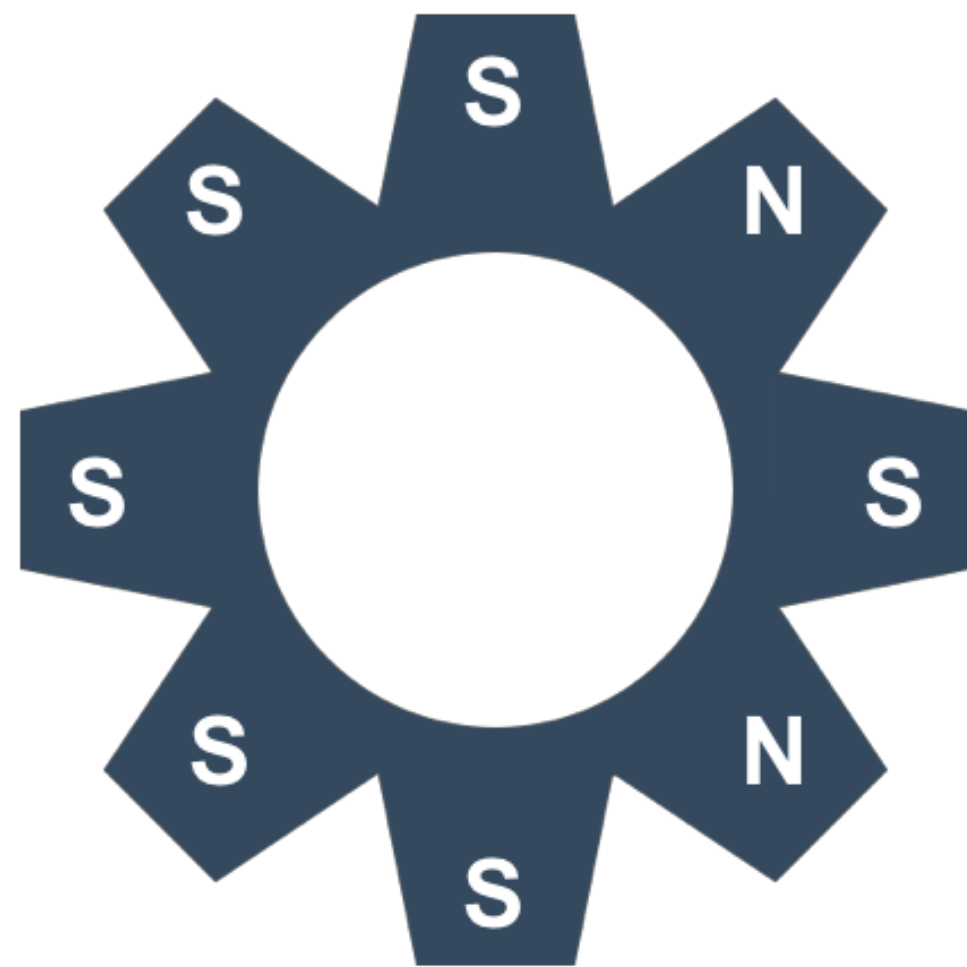
<https://www.acmicpc.net/problem/14499>

• 소스: <http://codeplus.codes/3304758f92d74f65af64bb4df4143608>

톱니바퀴

<https://www.acmicpc.net/problem/14890>

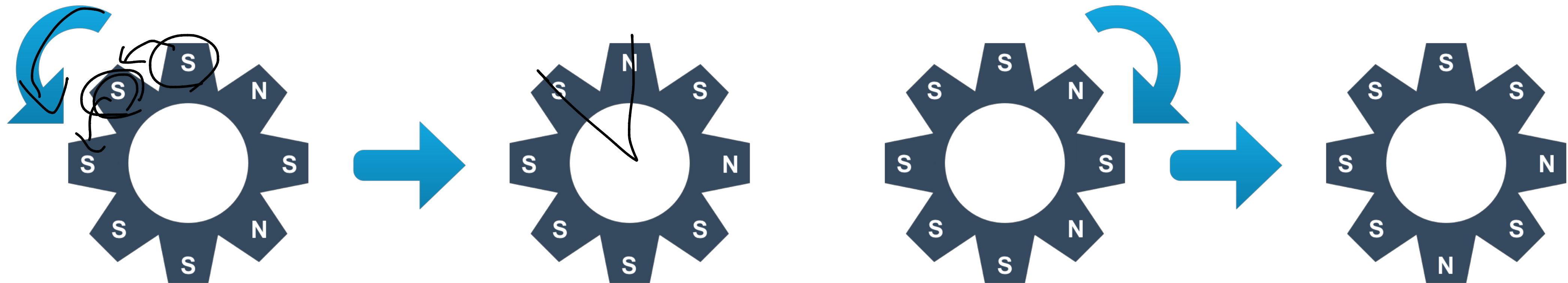
- 총 8개의 톱니를 가지고 있는 톱니바퀴 4개가 일렬로 놓여져 있다.



톱니바퀴

<https://www.acmicpc.net/problem/14890>

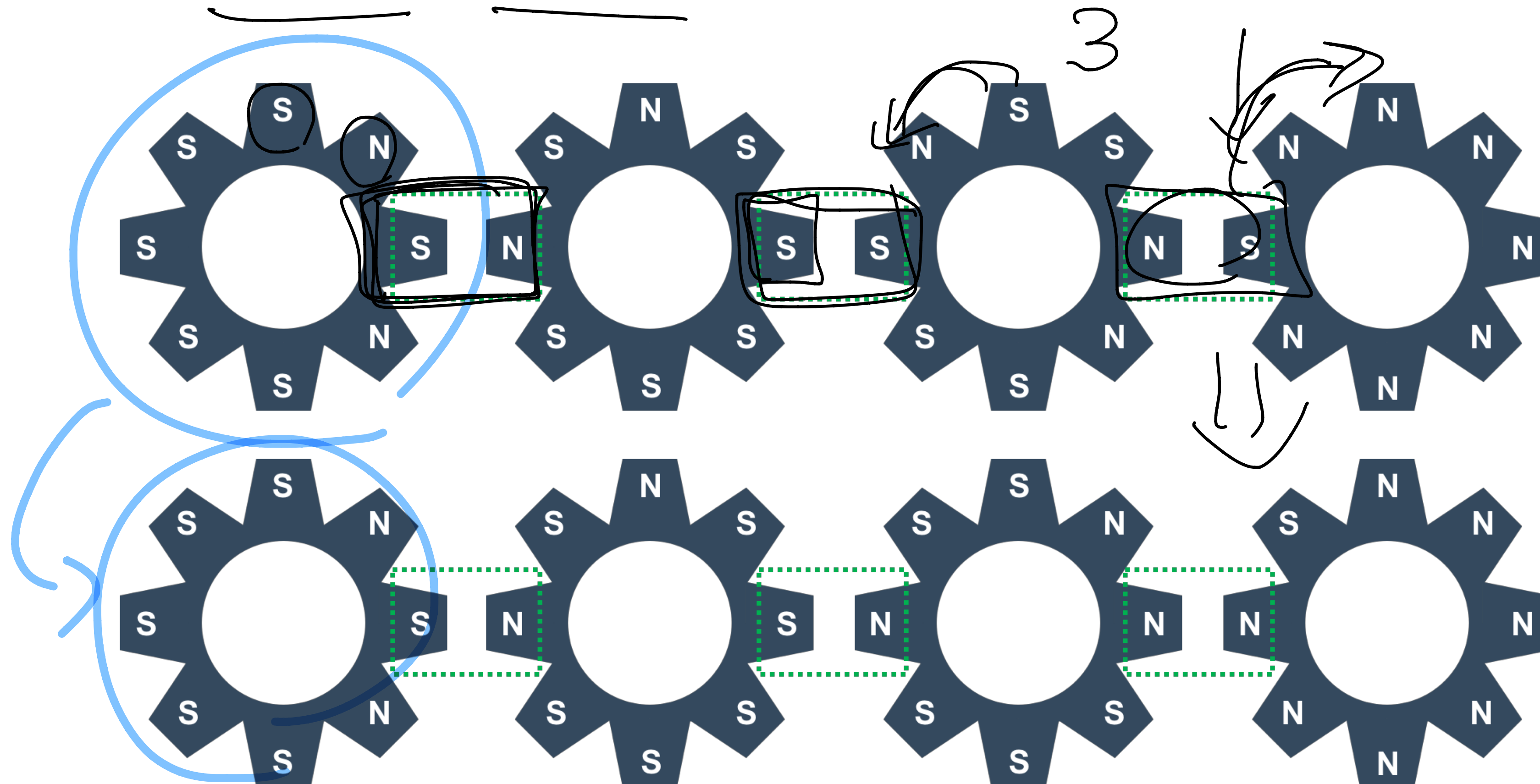
- 톱니바퀴를 총 K번 이동시키려고 한다. 회전은 시계 방향과 반시계 방향이 있다.



톱니바퀴

<https://www.acmicpc.net/problem/14890>

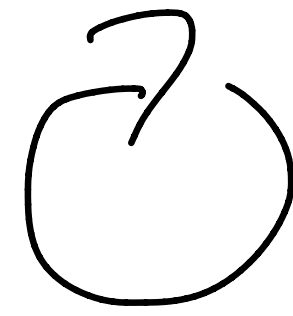
- 톱니바퀴 A를 회전할 때, 그 옆에 있는 톱니바퀴 B와 서로 맞닿은 톱니의 극이 다르다면, B는 A가 회전한 방향과 반대방향으로 회전하게 된다. (그림: 3번을 반시계 방향으로)



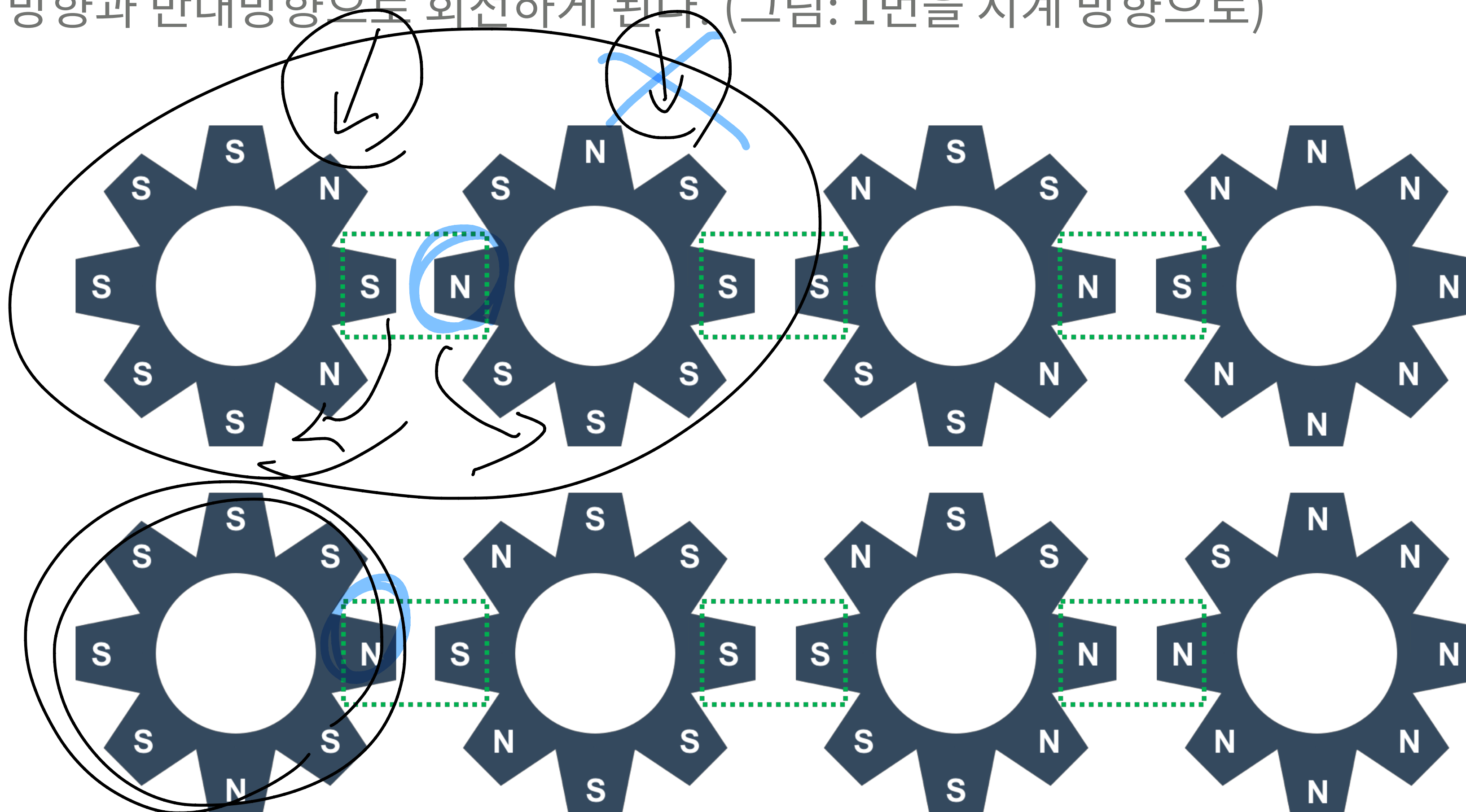
톱니바퀴

<https://www.acmicpc.net/problem/14890>

12



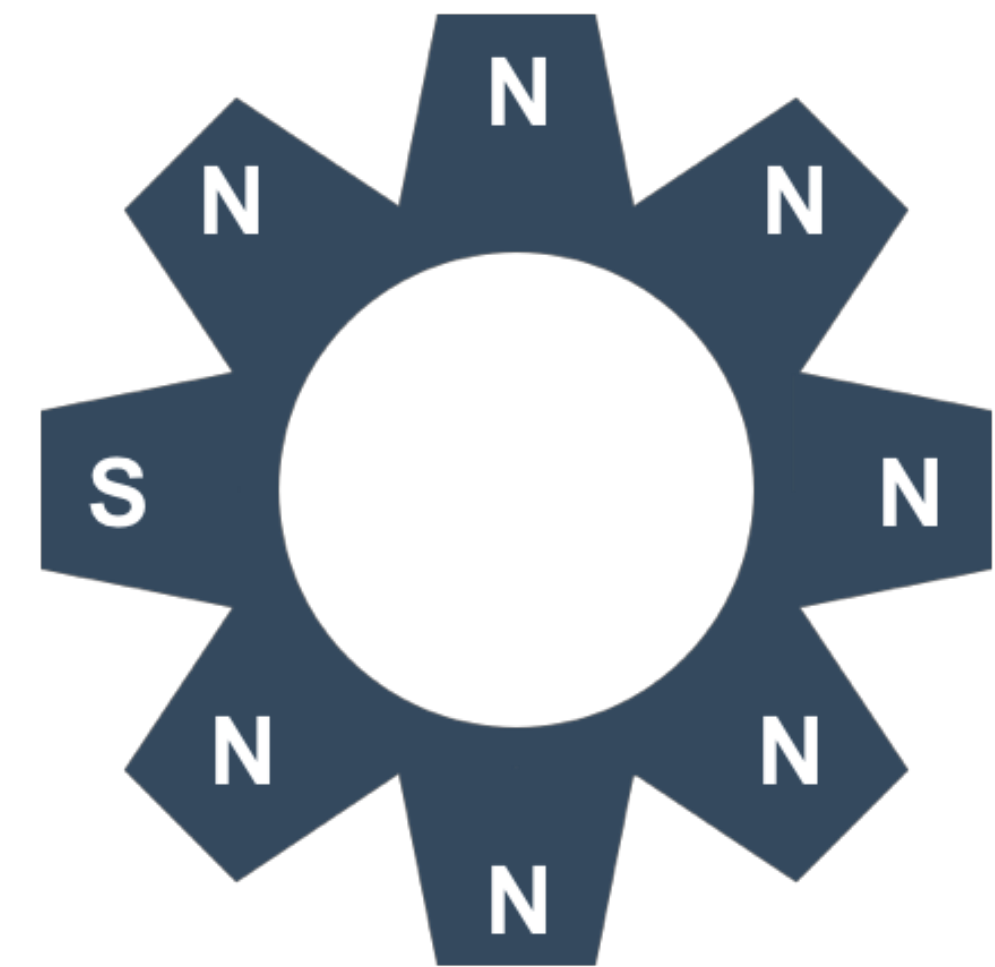
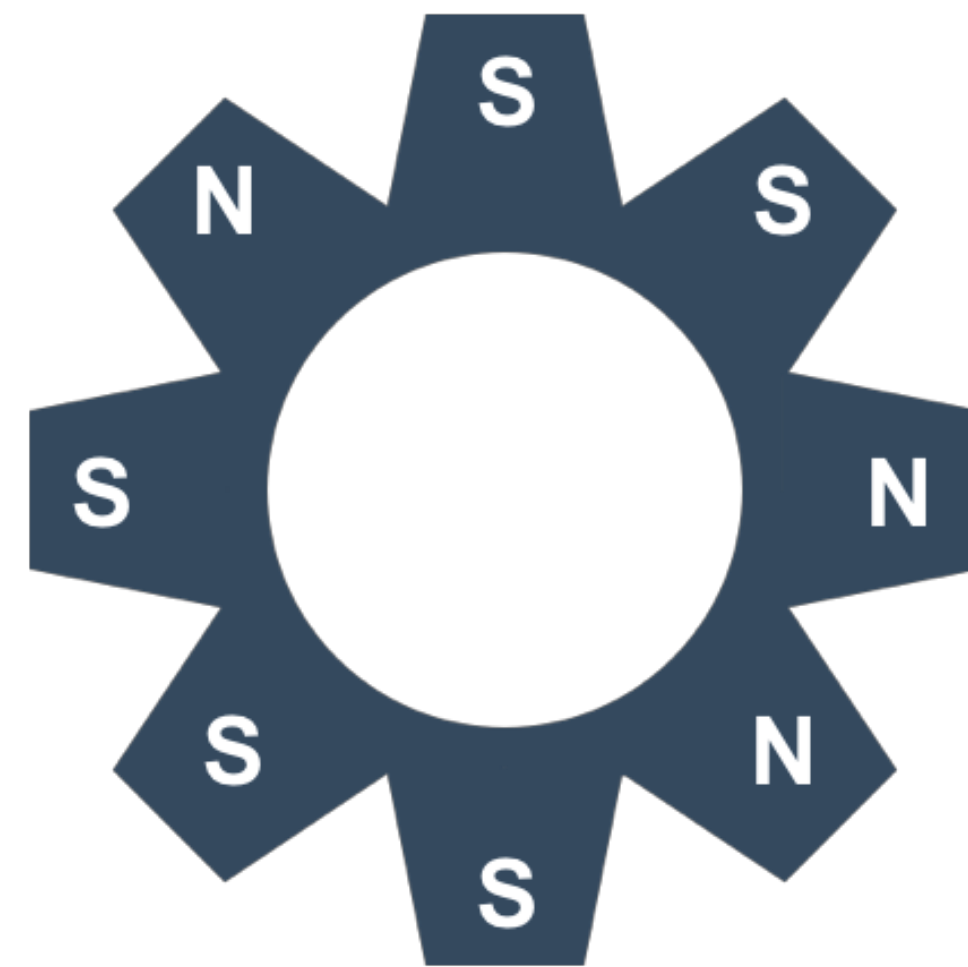
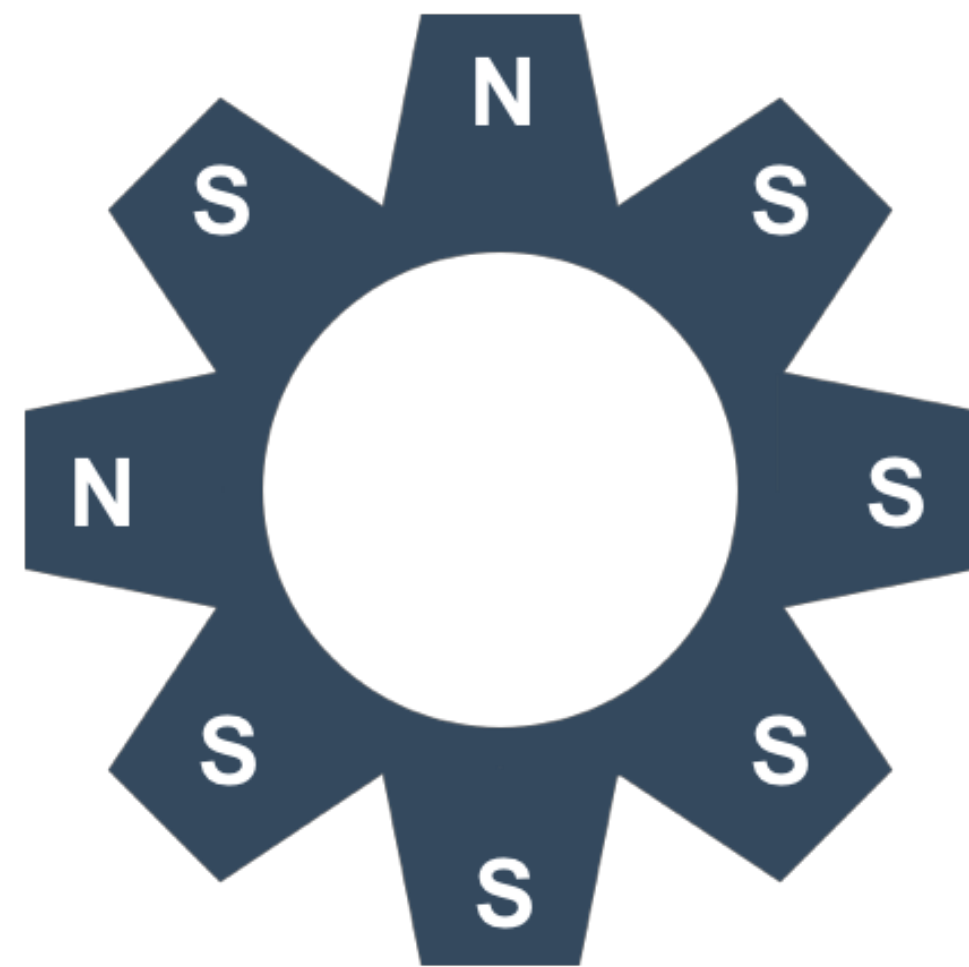
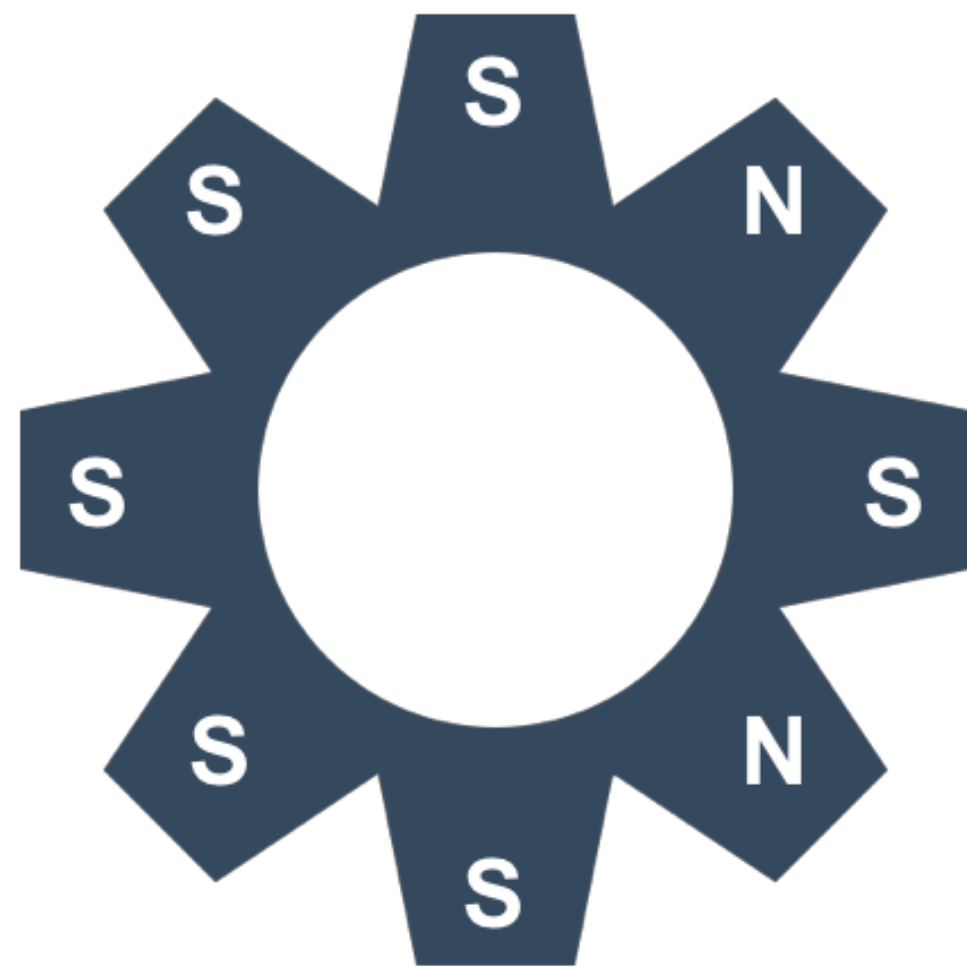
- 톱니바퀴 A를 회전할 때, 그 옆에 있는 톱니바퀴 B와 서로 맞닿은 톱니의 극이 다르다면, B는 A가 회전한 방향과 반대방향으로 회전하게 된다 (그림: 1번을 시계 방향으로)



톱니바퀴

<https://www.acmicpc.net/problem/14890>

- 문제에 나와있는 대로 구현을 해야 한다.



톱니바퀴

<https://www.acmicpc.net/problem/14890>

- 소스: <http://codeplus.codes/094a42a9f1ad48afb65ecdcd638a3f5c>

톱니바퀴 2

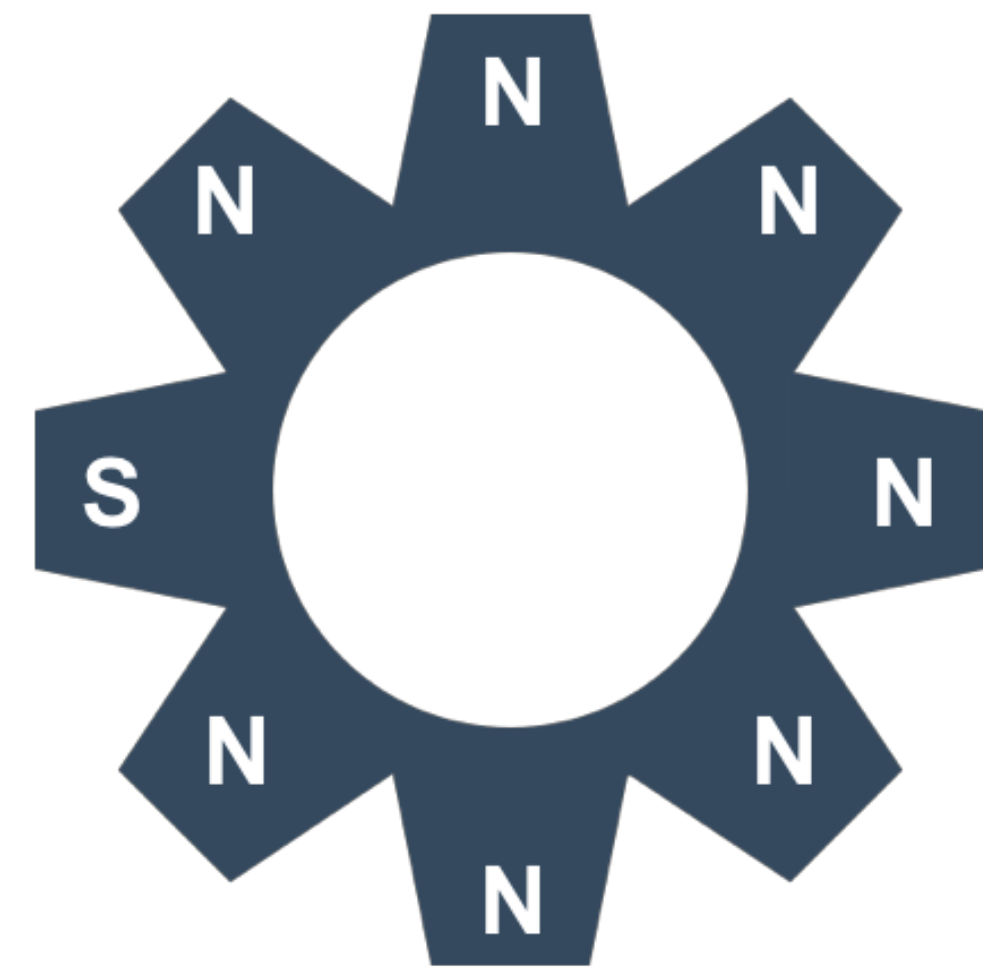
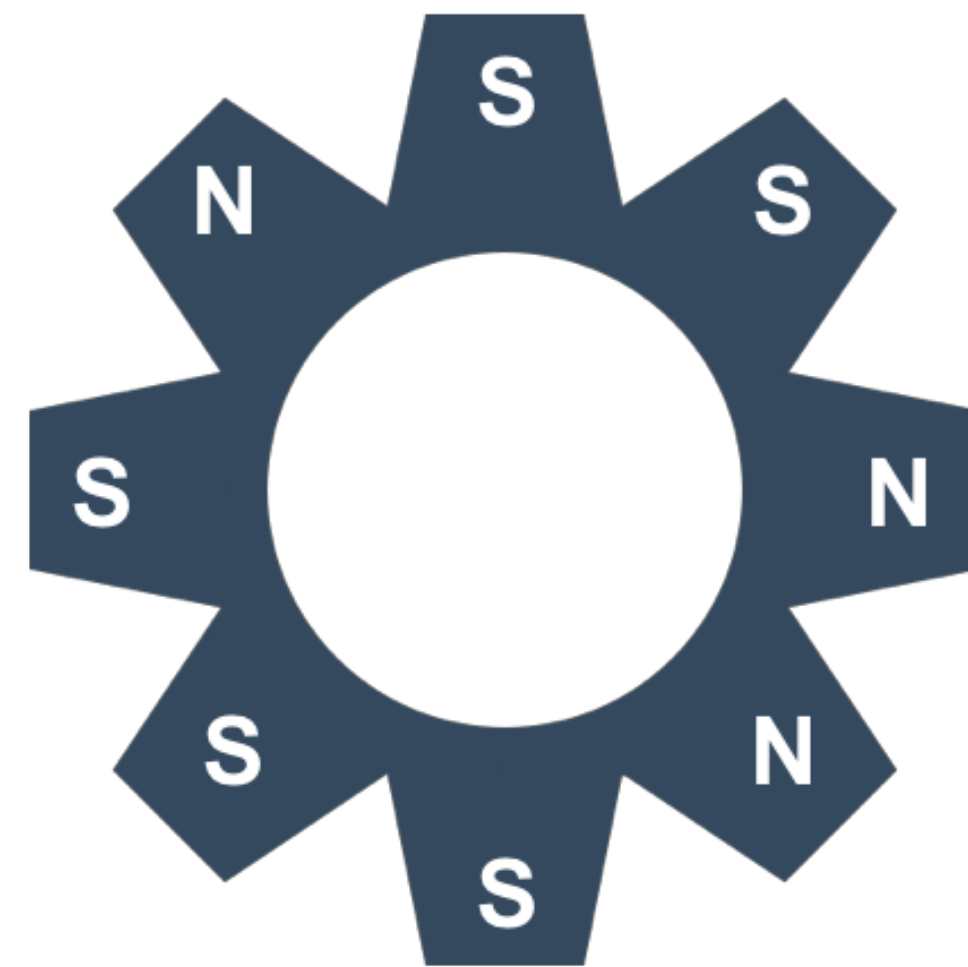
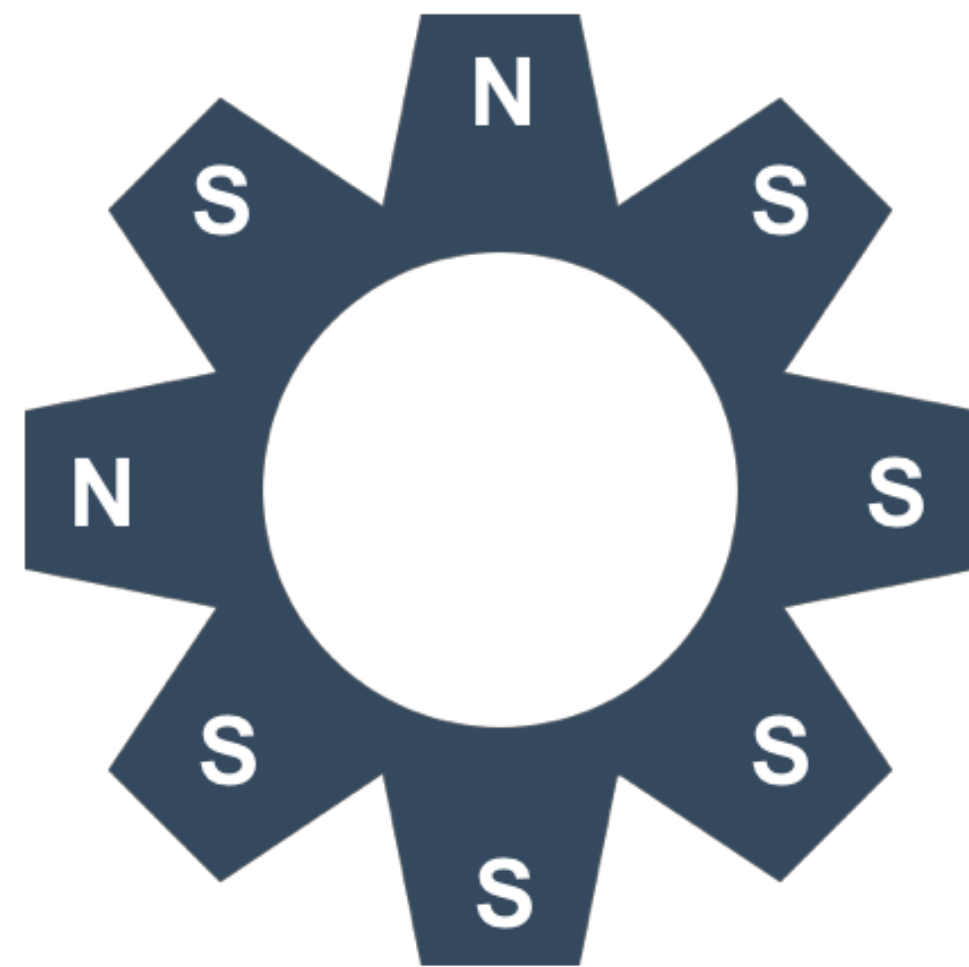
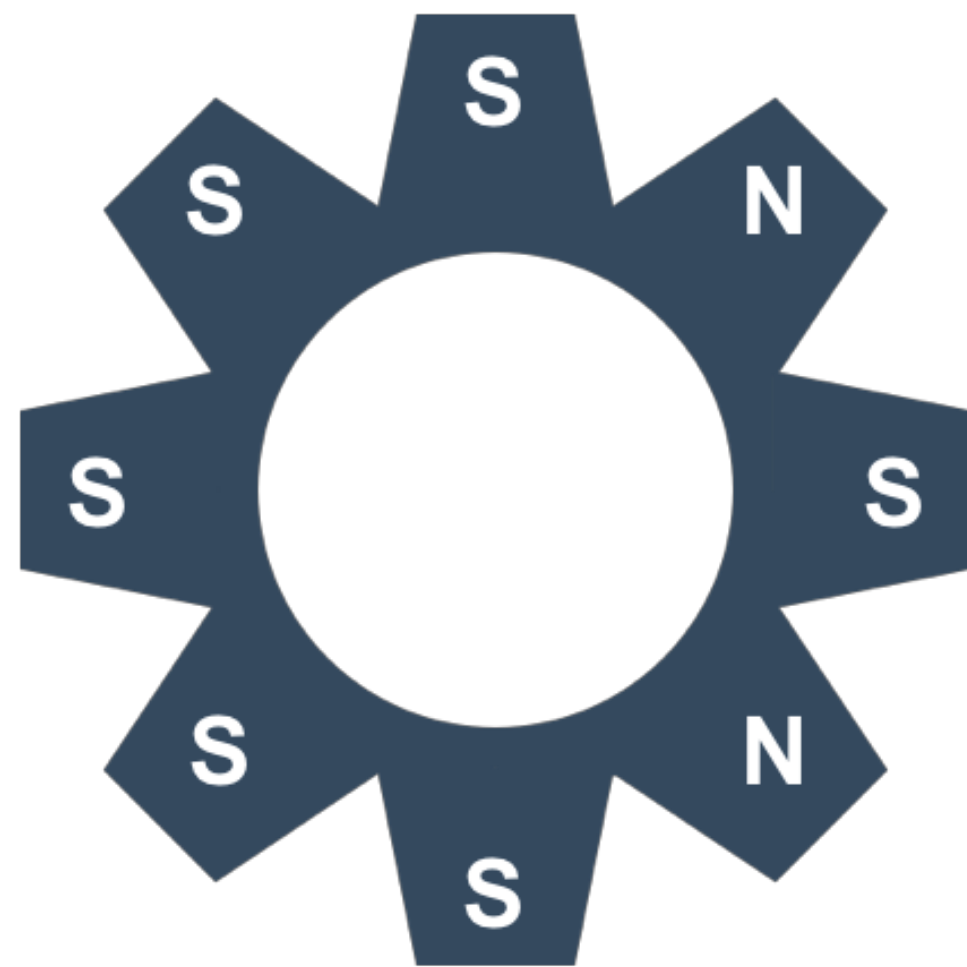
<https://www.acmicpc.net/problem/15662>

15

- 총 8개의 톱니를 가지고 있는 톱니바퀴 N개가 일렬로 놓여져 있다.

16
4

① $f(2)$
 $f(3)$
 $f(4)$



톱니바퀴 2

<https://www.acmicpc.net/problem/15662>

- 톱니바퀴와 다른 점이 톱니바퀴의 개수이기 때문에, 같은 소스로 풀 수 있다.

톱니바퀴 2

<https://www.acmicpc.net/problem/15662>

- 소스: <http://codeplus.codes/9070a0024f1342268c871b1a85b189a5>

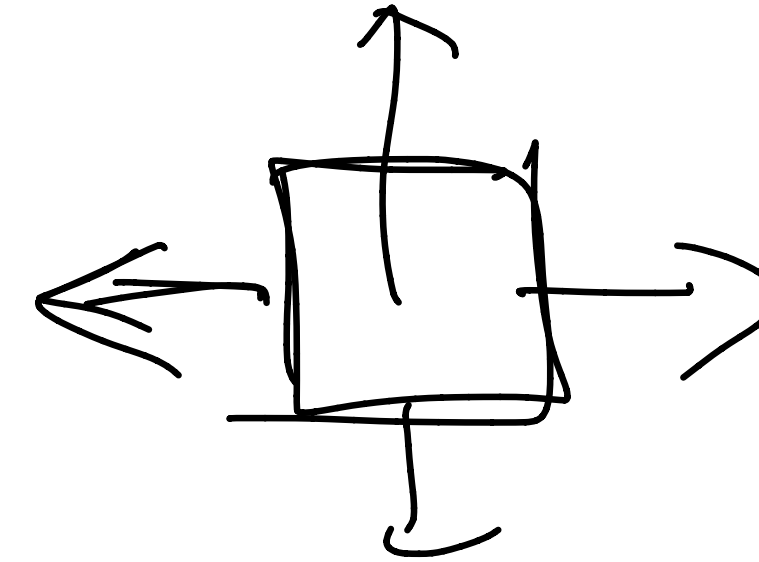
로봇 청소기

<https://www.acmicpc.net/problem/14503>

- 크기가 $N \times M$ 인 지도 위의 한 칸에 로봇 청소기가 놓여져 있다.
- 문제에 주어진 대로 로봇 청소기를 작동시켜서, 총 몇 개의 칸을 청소하는지 구하는 문제

로봇 청소기

<https://www.acmicpc.net/problem/14503>



1. 현재 위치를 청소한다.

2. 현재 위치에서 현재 방향을 기준으로 왼쪽방향부터 차례대로 탐색을 진행한다.

1. 왼쪽 방향에 아직 청소하지 않은 공간이 존재한다면, 그 방향으로 회전한 다음 한 칸을 전진하고 1번부터 진행한다.

2. 왼쪽 방향에 청소할 공간이 없다면, 그 방향으로 회전하고 2번으로 돌아간다.

3. 네 방향 모두 청소가 이미 되어있거나 벽인 경우에는, 바라보는 방향을 유지한 채로 한 칸 후진을 하고 2번으로 돌아간다.

4. 네 방향 모두 청소가 이미 되어있거나 벽이면서, 뒤쪽 방향이 벽이라 후진도 할 수 없는 경우에는 작동을 멈춘다.

로봇 청소기

<https://www.acmicpc.net/problem/14503>

- 문제에 주어진 대로 구현해야 한다.

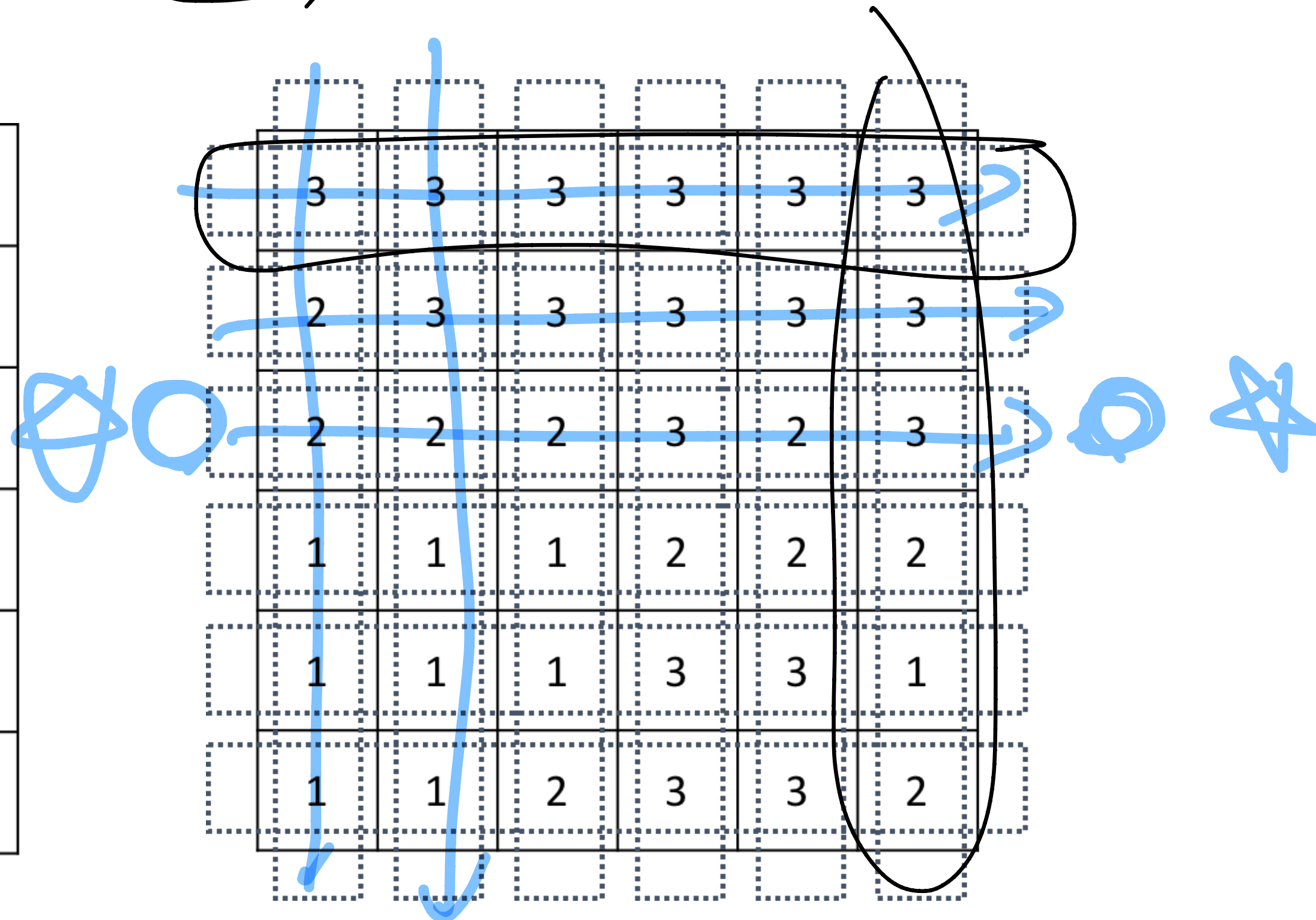
로봇 청소기

<https://www.acmicpc.net/problem/14503>

- 소스: <http://codeplus.codes/bd02caedbe774a3f9fa0be38c38f3e33>

- $N \times N$ 크기의 지도가 있고, 각 칸에는 높이가 적혀져 있다.
- 길이란 한 행 또는 한 열 전체를 의미하고, 한쪽 끝에서 다른쪽 끝까지 지나가는 것

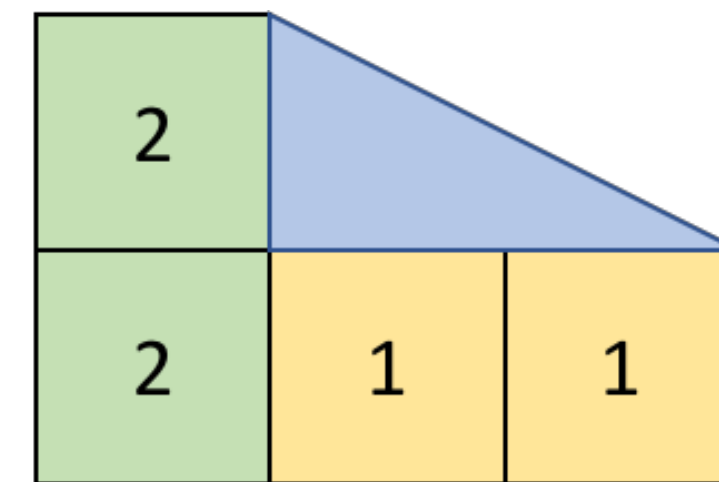
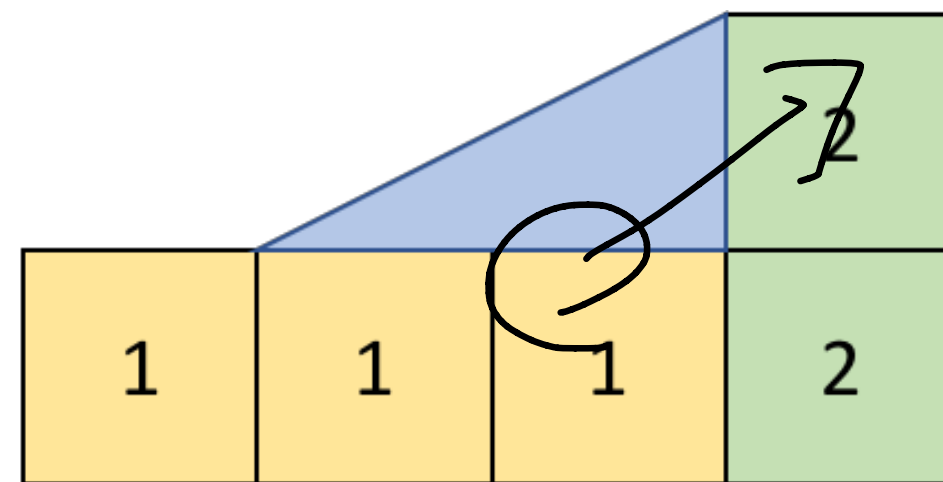
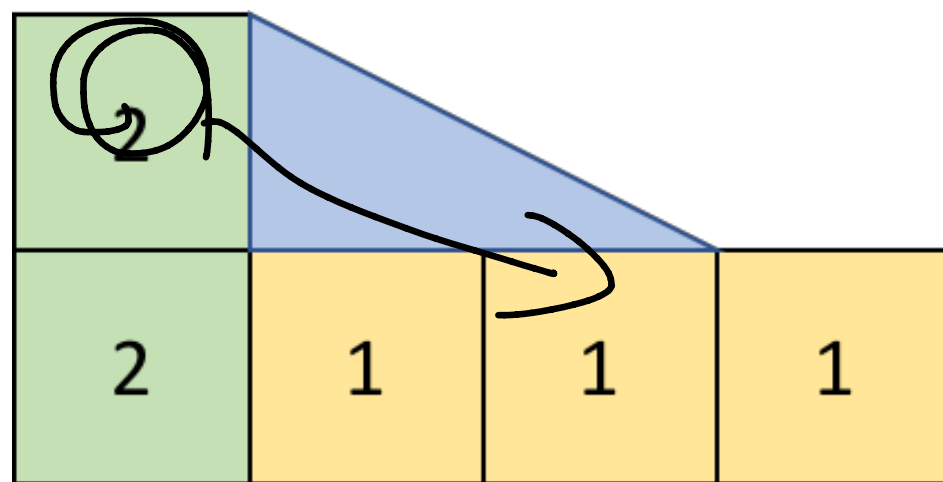
3	3	3	3	3	3
2	3	3	3	3	3
2	2	2	3	2	3
1	1	1	2	2	2
1	1	1	3	3	1
1	1	2	3	3	2



경사로

<https://www.acmicpc.net/problem/14890>

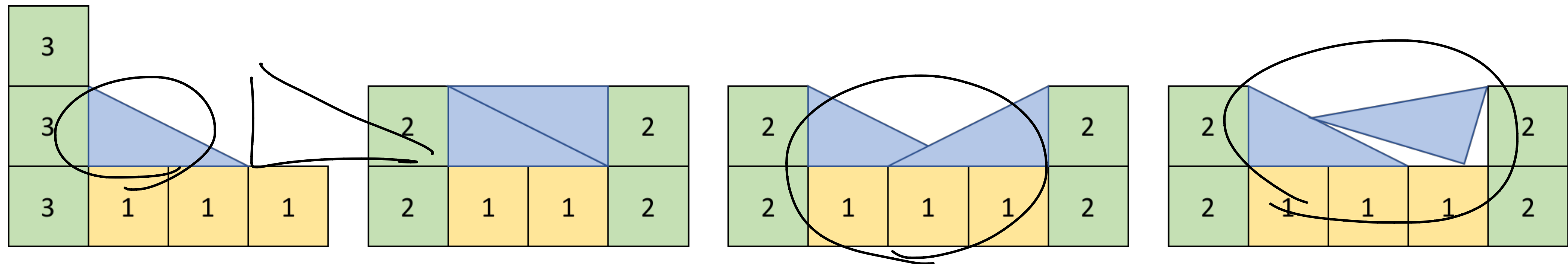
- 길에 속한 모든 칸의 높이가 같아야 지나갈 수 있다.
- 높이가 다른 경우에는 경사로를 놓아야 하며, 높이는 항상 1, 길이는 L이다.
 - 경사로는 낮은 칸에 놓으며, L개의 연속된 칸에 경사로의 바닥이 모두 접해야 한다.
 - 낮은 칸과 높은 칸의 높이 차이는 1이어야 한다.
 - 경사로를 놓을 낮은 칸의 높이는 모두 같아야 하고, L개의 칸이 연속되어 있어야 한다.



경사로

<https://www.acmicpc.net/problem/14890>

- 경사를 놓을 수 없는 경우는 다음과 같다.
 - 경사를 놓은 곳에 또 경사를 놓는 경우
 - 낮은 칸과 높은 칸의 높이 차이가 1이 아닌 경우
 - 낮은 지점의 칸의 높이가 모두 같지 않거나, L개가 연속되지 않은 경우
 - 경사를 놓다가 범위를 벗어나는 경우



경사로

<https://www.acmicpc.net/problem/14890>

25

길이 $2N$ 개

- $N \times N$ 크기의 지도가 있고, 각 칸에는 높이가 적혀져 있다.
- 길이란 한 행 또는 한 열 전체를 의미하고, 한쪽 끝에서 다른쪽 끝까지 지나가는 것

3	3	3	3	3	3
2	3	3	3	3	3
2	2	2	3	2	3
1	1	1	2	2	2
1	1	1	3	3	1
1	1	2	3	3	2

	3	3	3	3	3	3
	2	3	3	3	3	3
	2	2	2	3	2	3
	1	1	1	2	2	2
	1	1	1	3	3	1
	1	1	2	3	3	2

	3	3	3	3	3	3
	2	3	3	3	3	3
	2	2	2	3	2	3
	1	1	1	2	2	2
	1	1	1	3	3	1
	1	1	2	3	3	2

경사로

<https://www.acmicpc.net/problem/14890>

- 이 문제도 문제에 나와있는 대로 구현을 해야 한다.

경사로

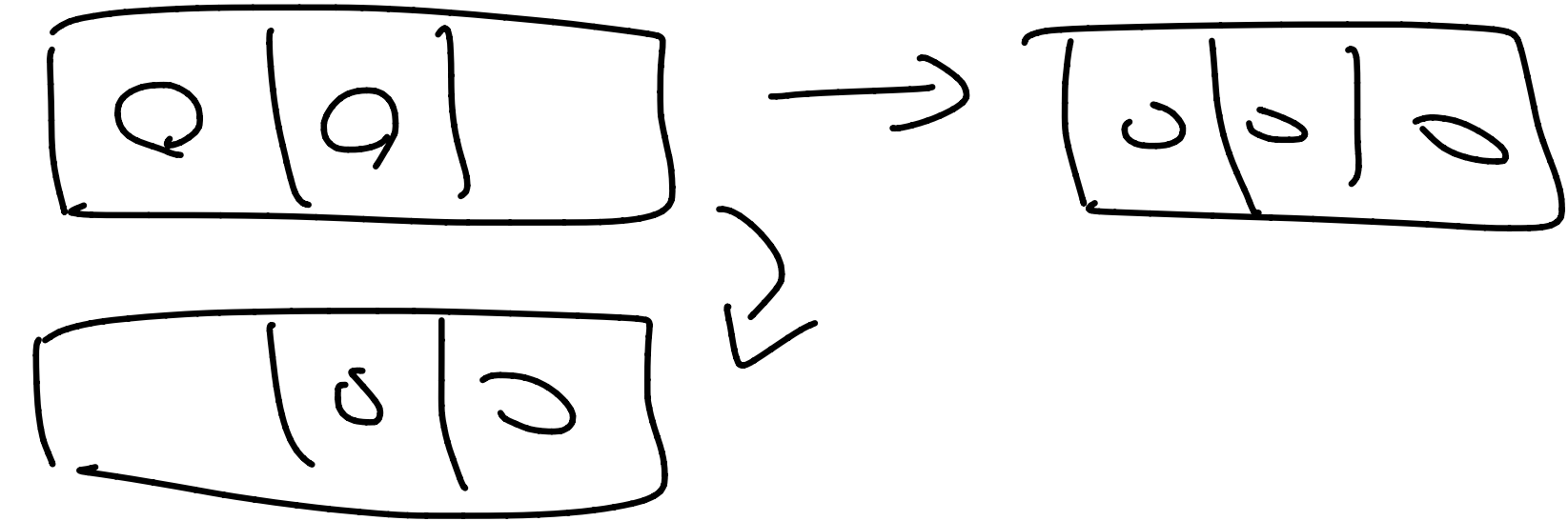
27

<https://www.acmicpc.net/problem/14890>

- 소스: <http://codeplus.codes/226fc8bf976d4aaca8618b48c50c105a>



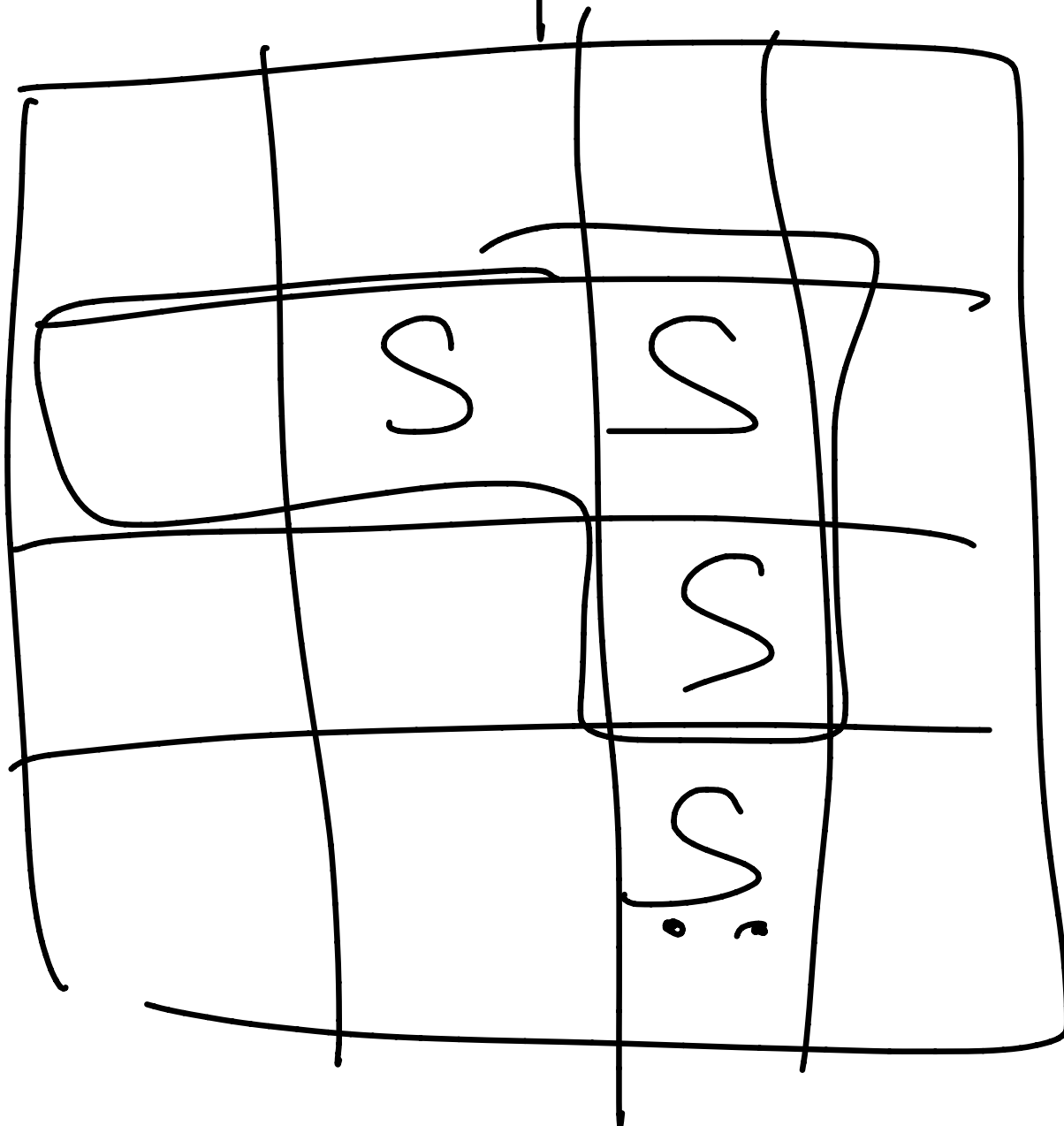
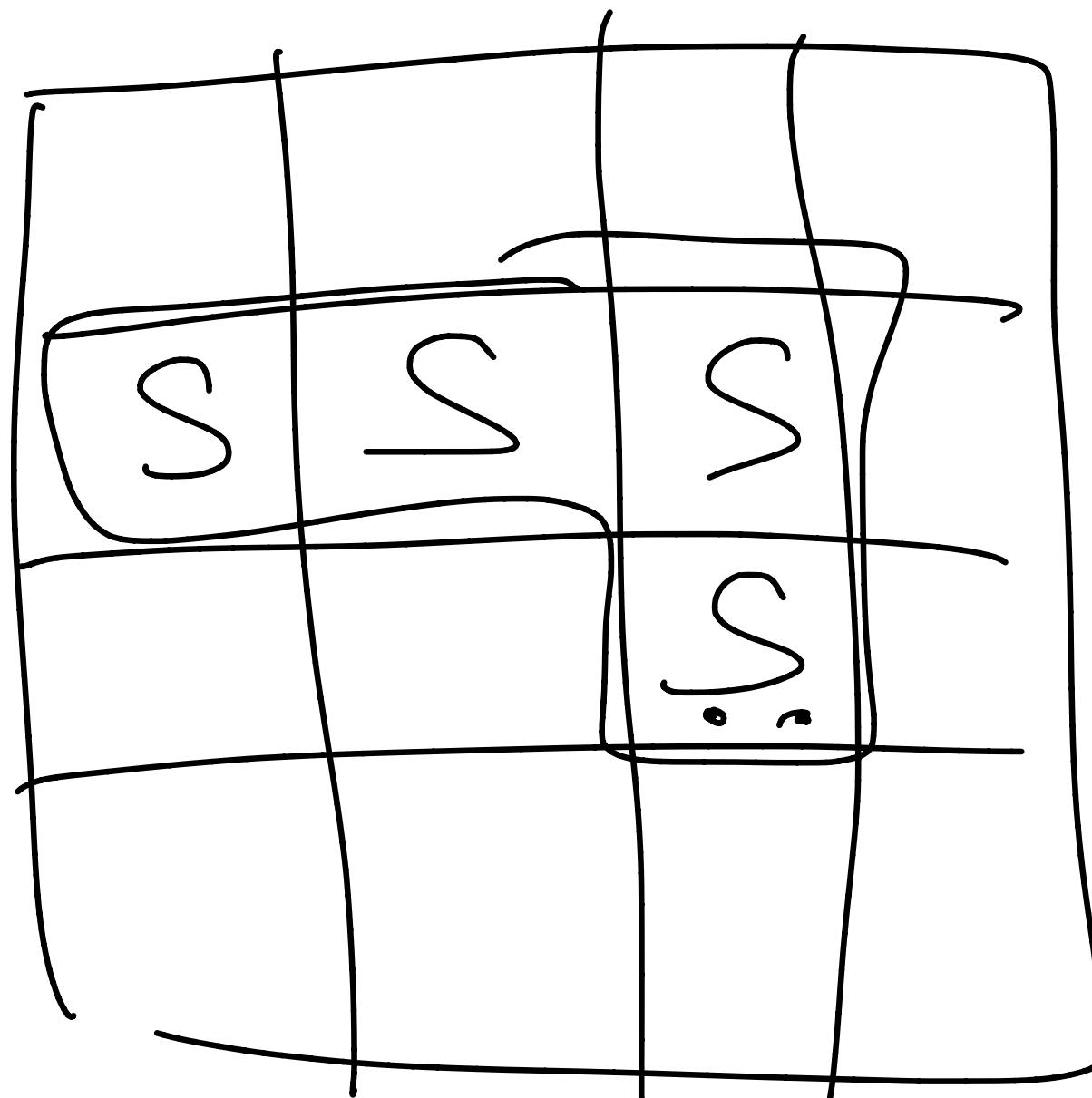
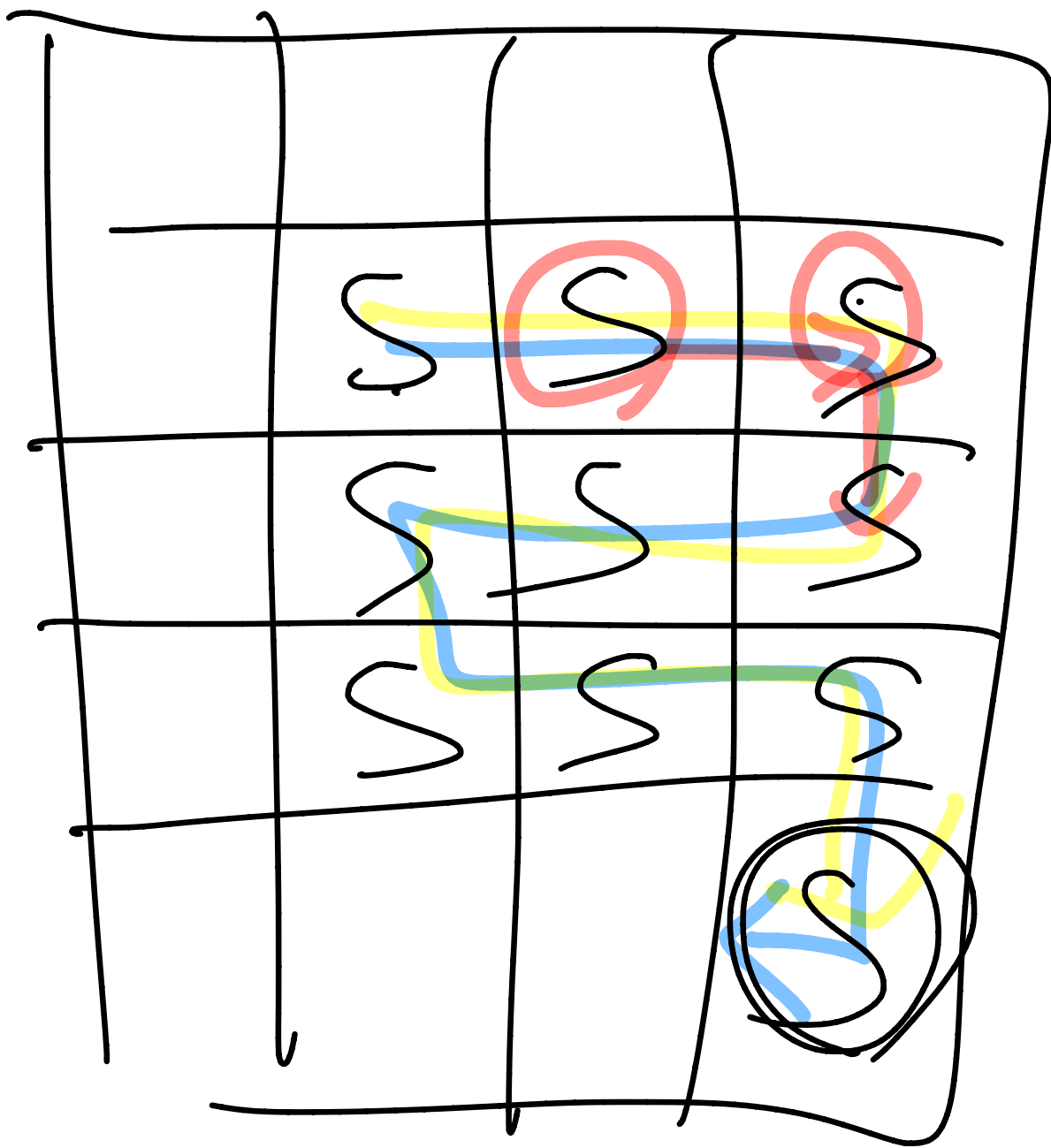
<https://www.acmicpc.net/problem/3190>



- $N \times N$ 크기의 보드 위에 뱀이 있고, 사과가 있다.
- 매 초마다 뱀은 다음과 같은 규칙에 따라서 이동한다.
 - 먼저 뱀은 몸길이를 늘려 머리를 다음칸에 위치시킨다.
 - 만약 이동한 칸에 사과가 있다면, 그 칸에 있던 사과가 없어지고 꼬리는 움직이지 않는다.
 - 만약 이동한 칸에 사과가 없다면, 몸길이를 줄여서 꼬리가 위치한 칸을 비워준다. 즉, 몸길이는 변하지 않는다.

<https://www.acmicpc.net/problem/3190>

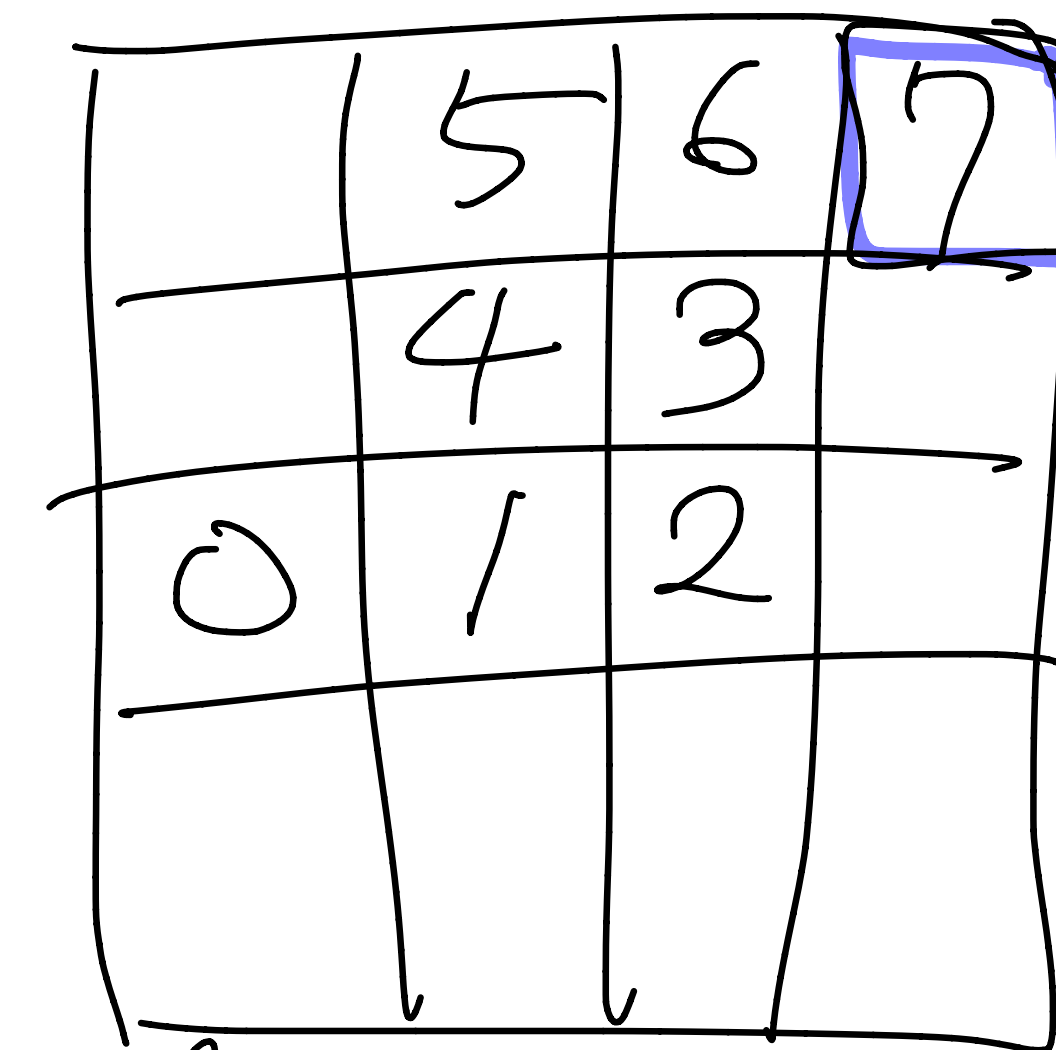
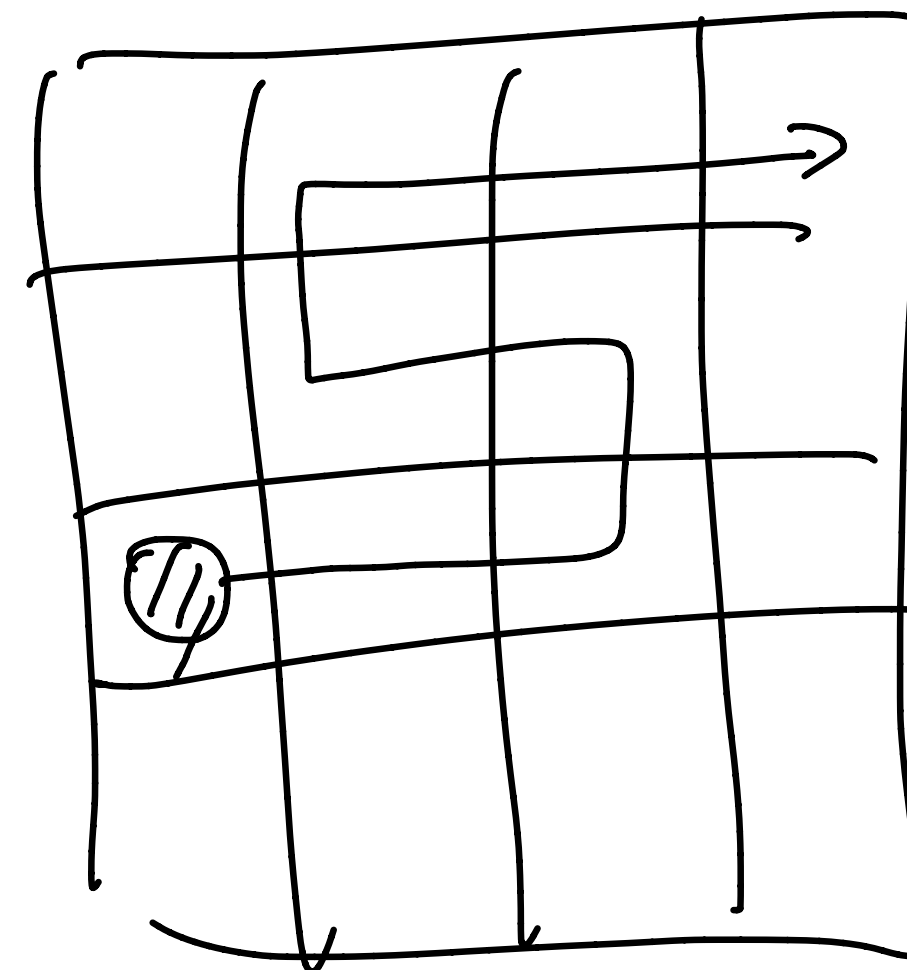
- 문제에 나와있는 대로 시뮬레이션 하면 된다.



뱀

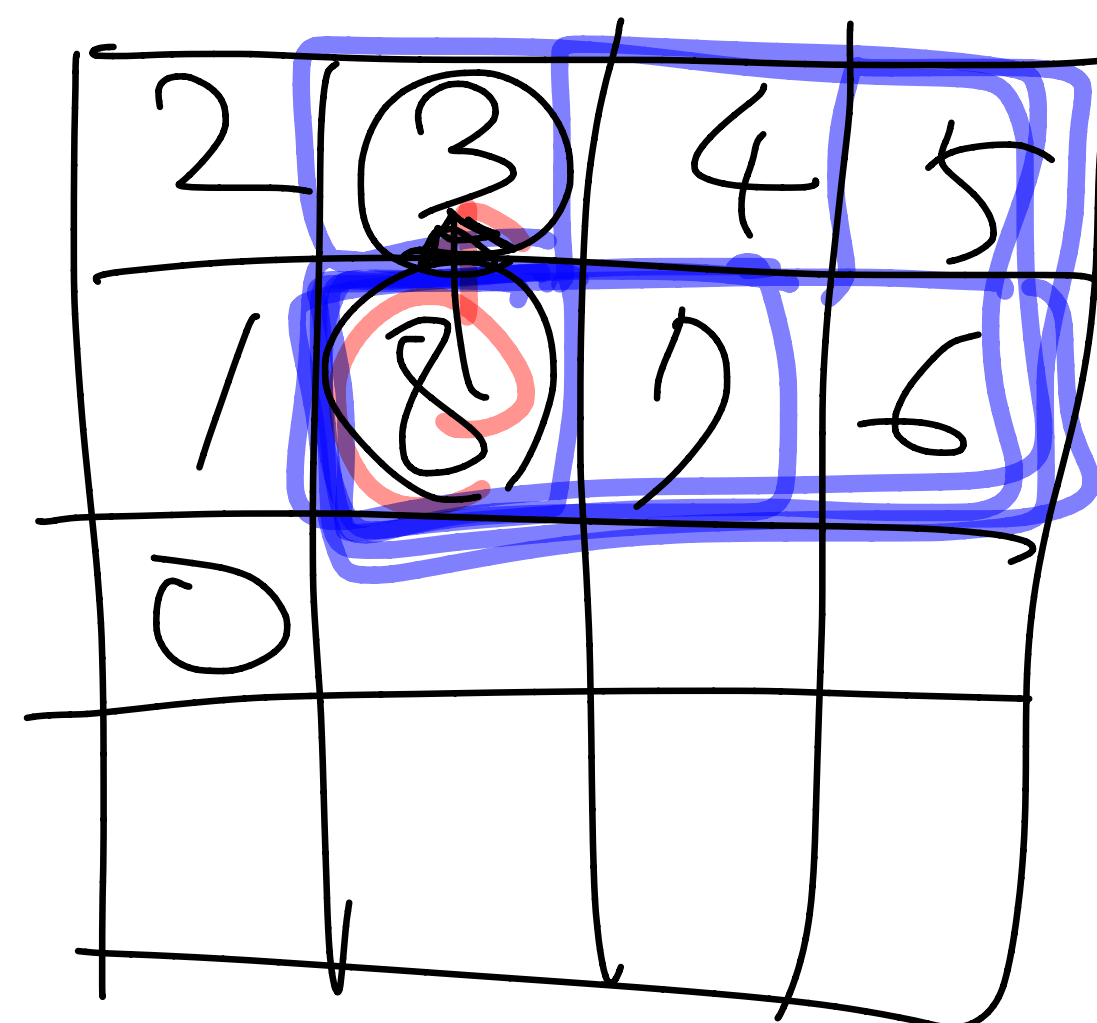
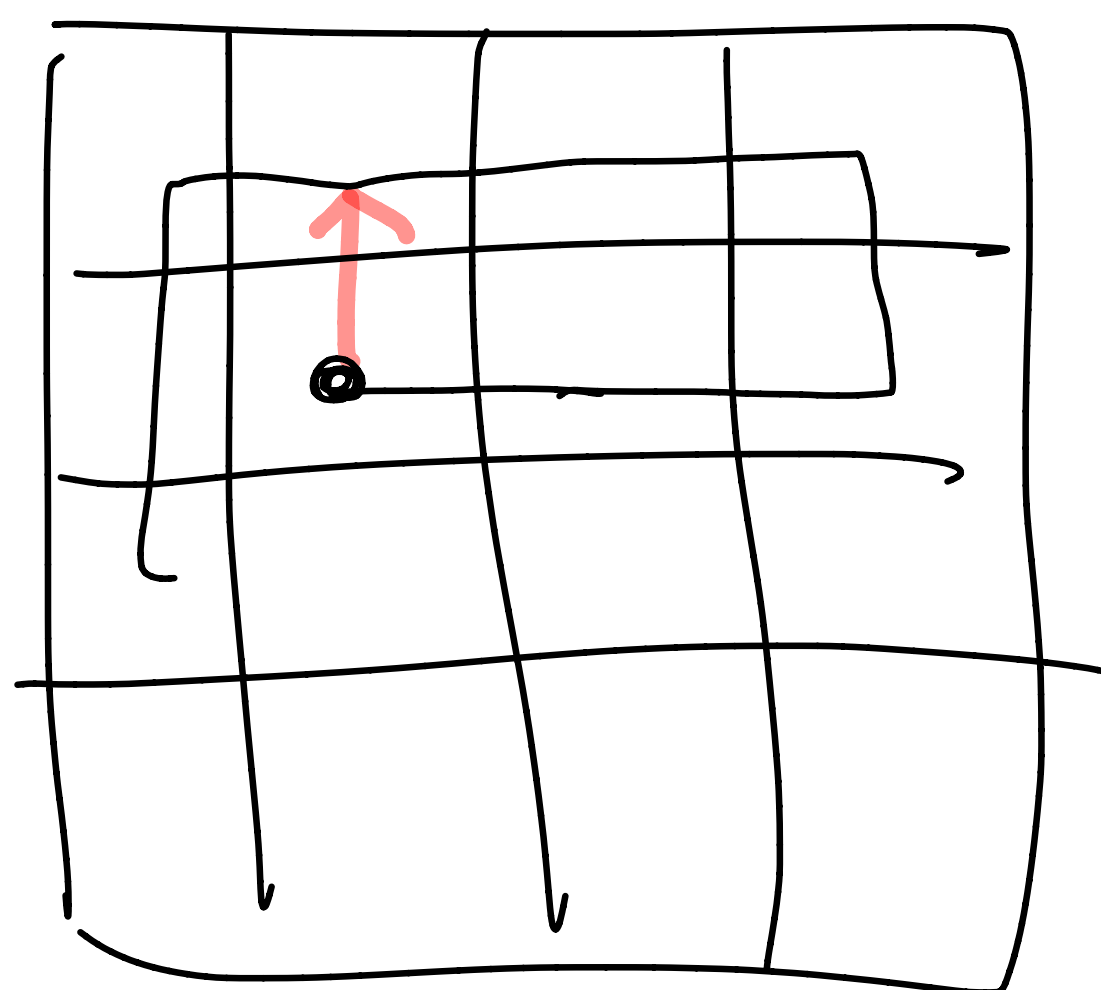
<https://www.acmicpc.net/problem/3190>

$\frac{1}{2} \times 2 = 1$



$D[r, c] - 1$ $D[r, c]$

- 뱀의 정보를 저장하는 것이 조금 어렵다.
- 뱀의 정보는 머리의 위치만 저장하면 된다.
- 머리를 제외한 나머지 칸은 머리를 쫓아가기 때문에
- 각각의 칸을 머리가 몇 초에 방문했는지 기록한다.
- 만약, 방문한 칸을 지나갈 때, 현재 시간과 그 칸을 방문한 시간의 차이를 길이와 비교한다.



5



<https://www.acmicpc.net/problem/3190>

- 소스: <http://codeplus.codes/31a47ba24b32436ca578e67dbe76d3da>

드래곤 커브

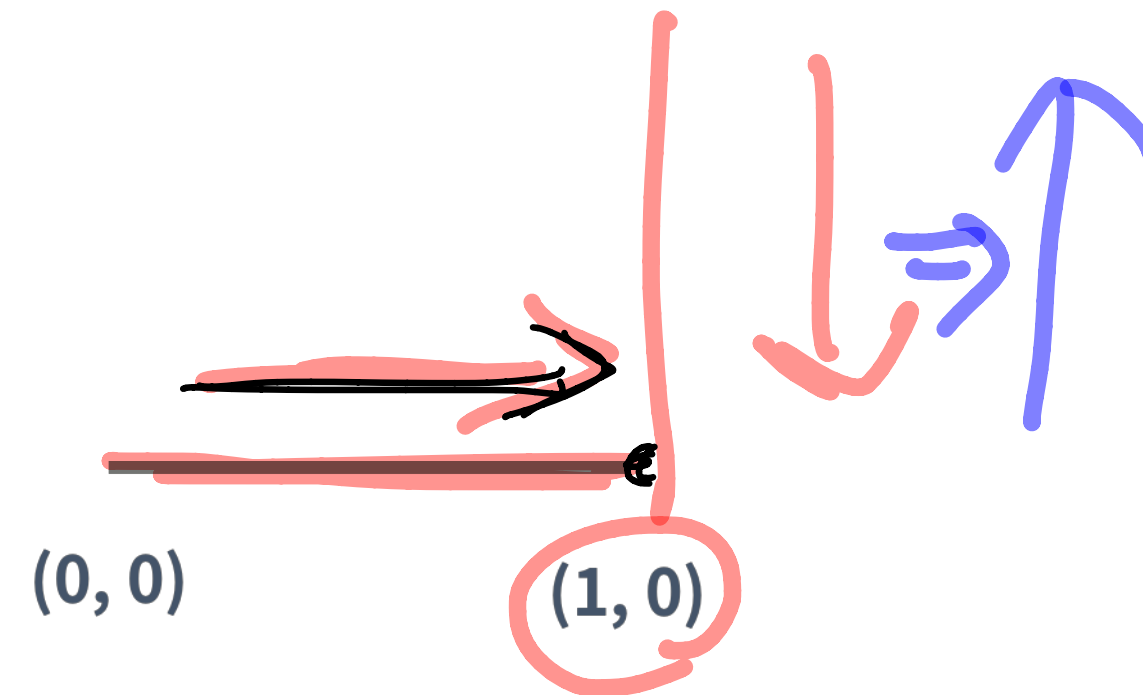
<https://www.acmicpc.net/problem/15685>

- 드래곤 커브는 세 가지 속성으로 이루어져 있다
 1. 시작 점
 2. 시작 방향
 3. 세대

드래곤 커브

<https://www.acmicpc.net/problem/15685>

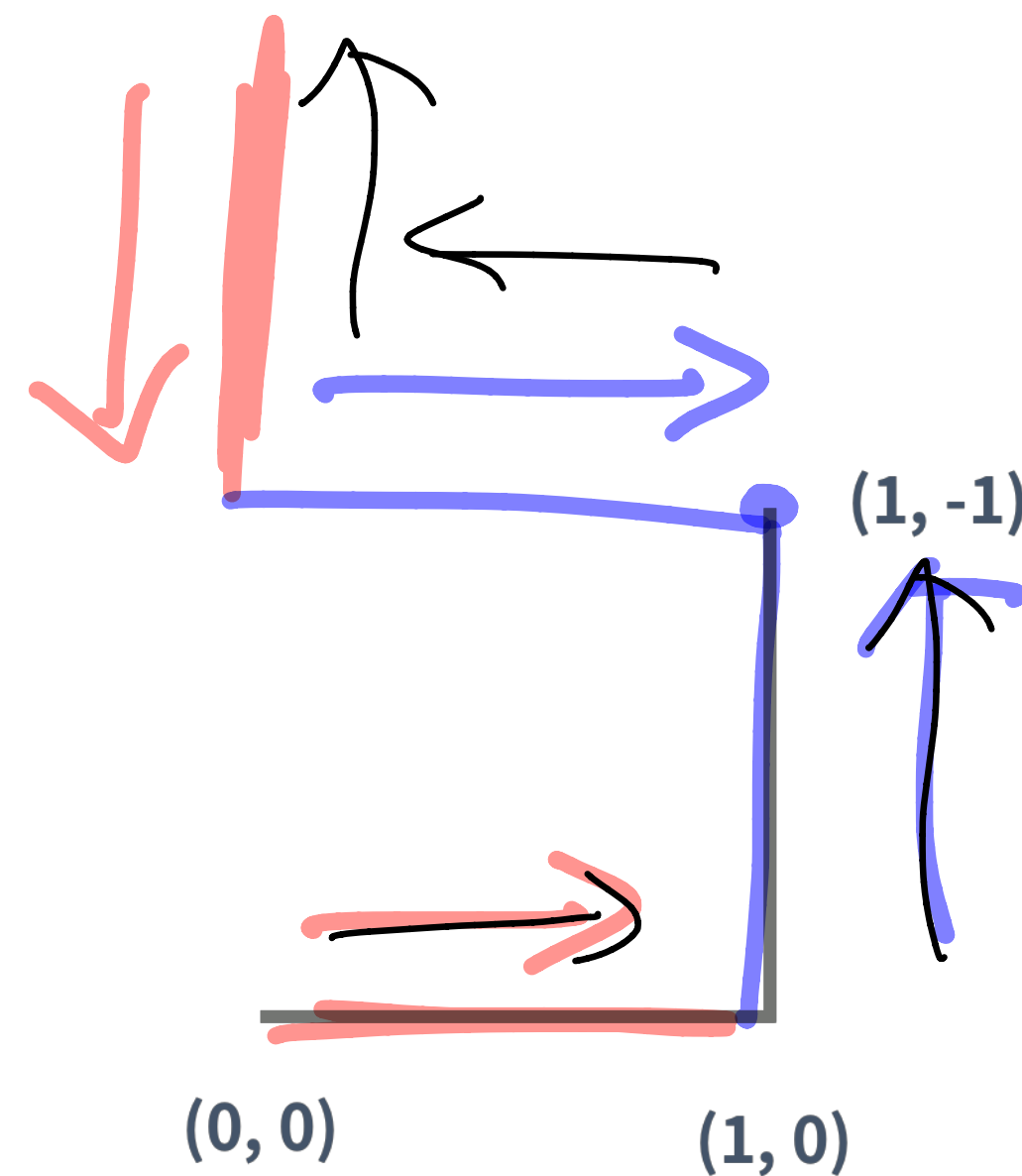
- $(0,0)$ 에서 오른쪽 방향으로 시작한 0세대 드래곤 커브



드래곤 커브

<https://www.acmicpc.net/problem/15685>

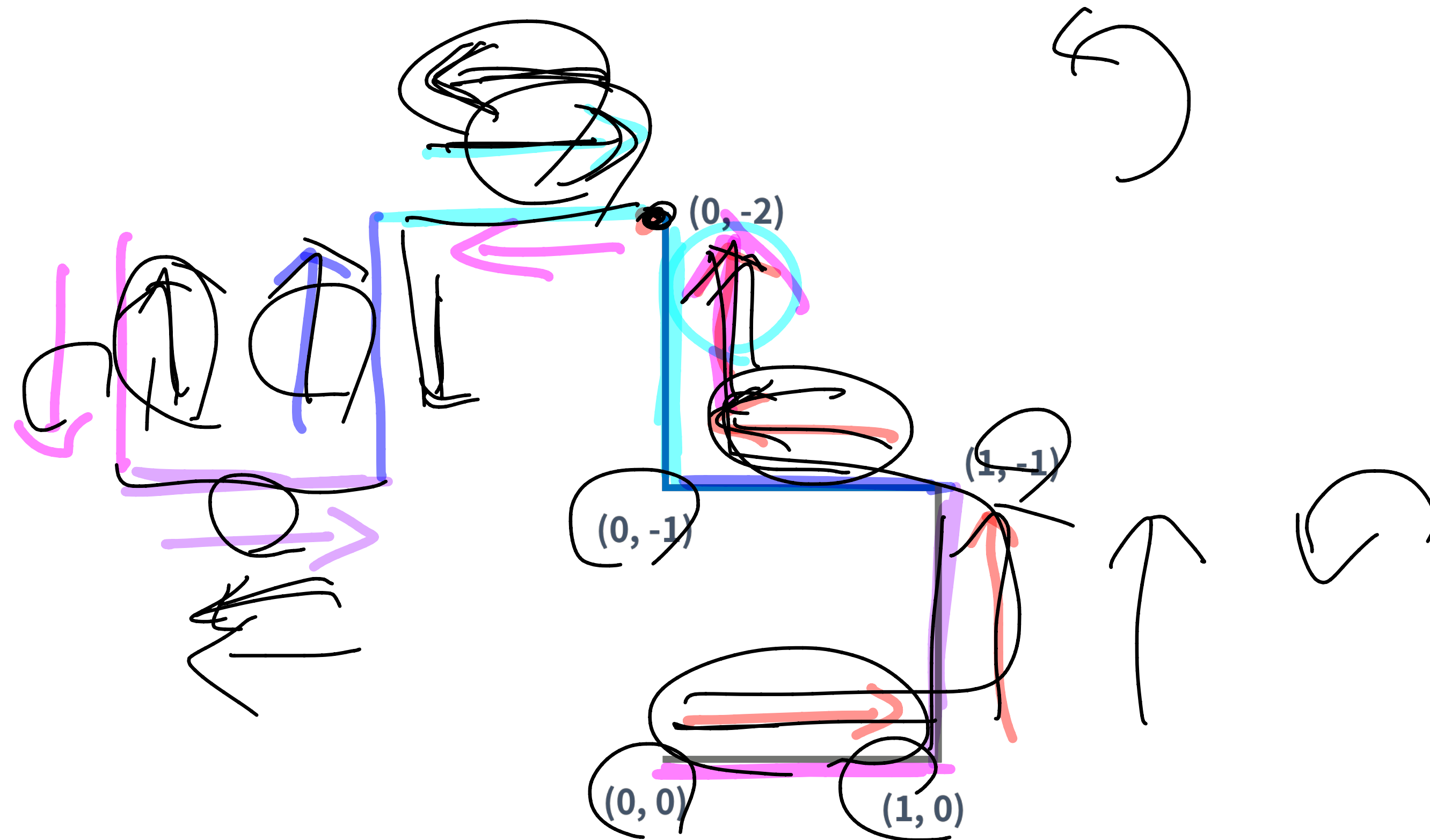
- $(0,0)$ 에서 오른쪽 방향으로 시작한 1세대 드래곤 커브
- 다음 세대는 이전 세대를 끝 점을 기준으로 90도 시계 방향 회전 시킨 다음 그 곳에 붙인 것



드래곤 커브

<https://www.acmicpc.net/problem/15685>

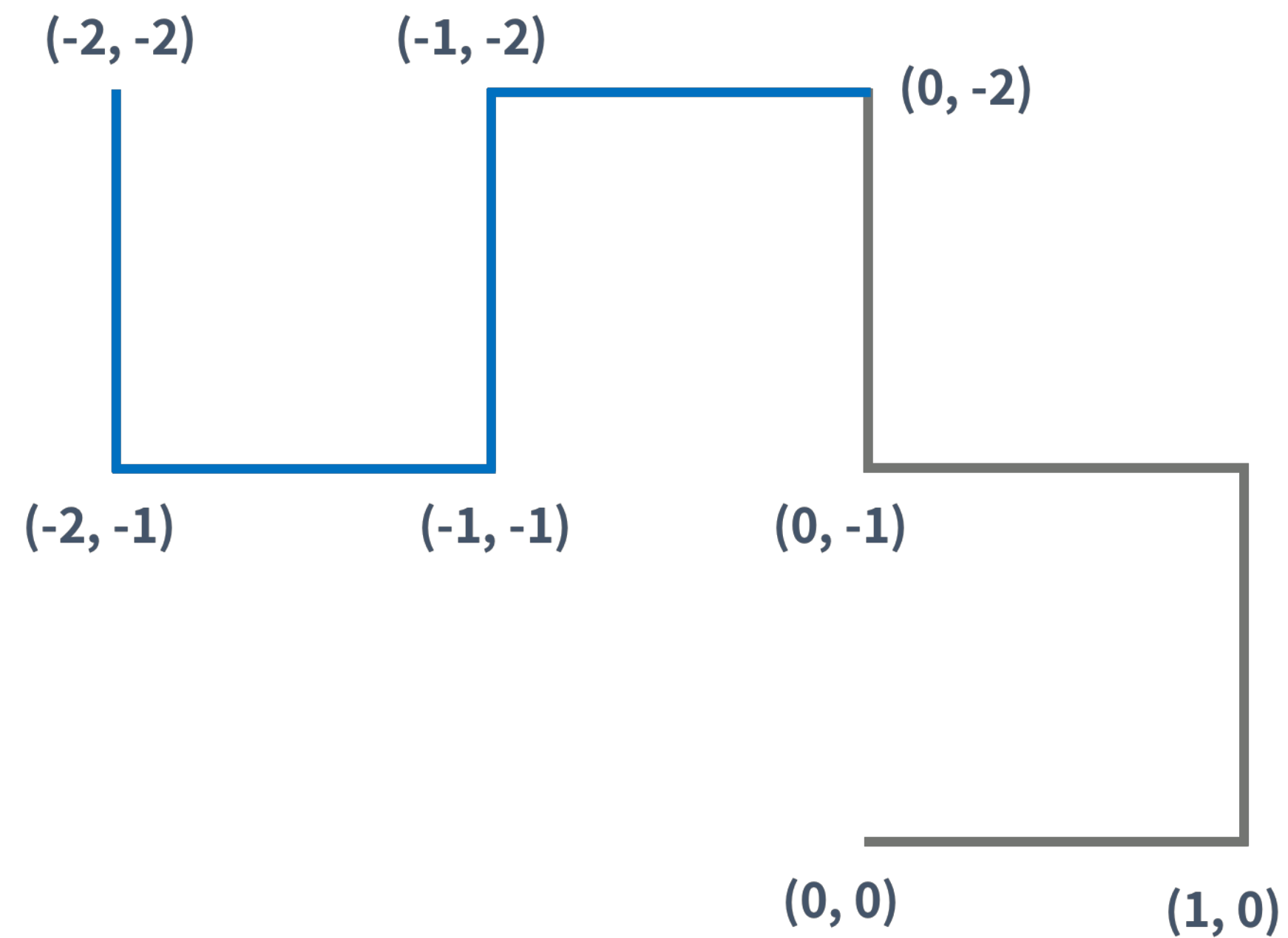
- $(0,0)$ 에서 오른쪽 방향으로 시작한 2세대 드래곤 커브
- 다음 세대는 이전 세대를 끝 점을 기준으로 90도 시계 방향 회전 시킨 다음 그 곳에 붙인 것



드래곤 커브

<https://www.acmicpc.net/problem/15685>

- $(0,0)$ 에서 오른쪽 방향으로 시작한 3세대 드래곤 커브
- 다음 세대는 이전 세대를 끝 점을 기준으로 90도 시계 방향 회전 시킨 다음 그 곳에 붙인 것



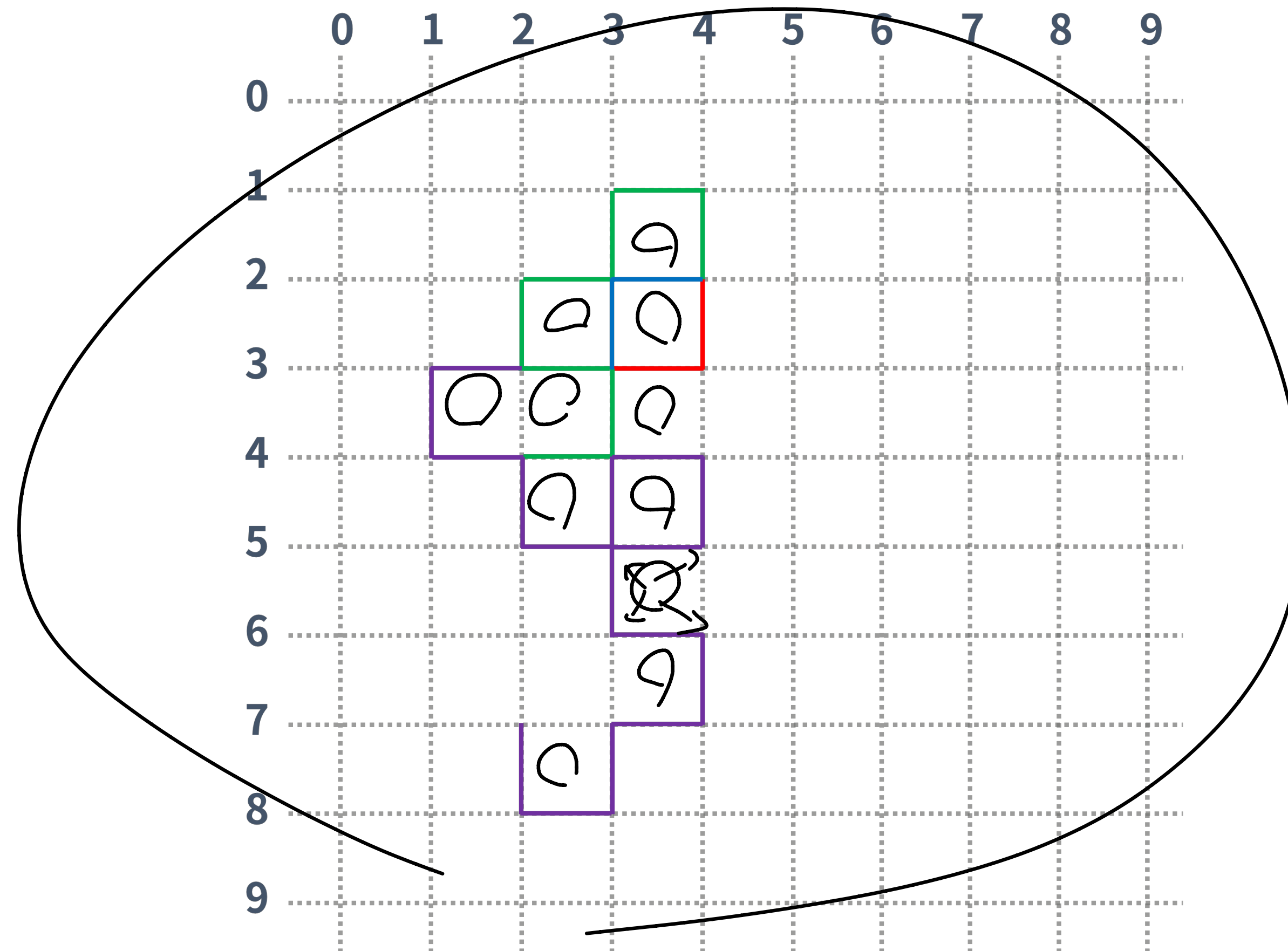
드래곤 커브

37

<https://www.acmicpc.net/problem/15685>

100 x 100

- 드래곤 커브가 N개 주어졌을 때, 격자의 네 꼭지점이 모두 드래곤 커브의 일부인 정사각형 개수를 구하는 문제



드래곤 커브

<https://www.acmicpc.net/problem/15685>

- 소스: <http://codeplus.codes/92dd9af3488a4660abb7f3d39e026dc8>

미네랄

<https://www.acmicpc.net/problem/2933>

- 동굴의 모양이 주어진다
- 두 사람은 던질 막대의 높이를 결정하고 막대를 던진다.
- 이 때, 미네랄을 만나면 미네랄은 모두 파괴되고, 막대는 그 위치에서 이동을 멈춘다.
- 새로운 클러스터가 공중에 떠있으면 중력에 의해서 아래로 내려온다.

미네랄

<https://www.acmicpc.net/problem/2933>

- 8)
- 7)
- ~~6)~~
- 5) . . . XXX . .
- 4) . . XXX . . .
- 3) . . X . XXX .
- 2) . . X . . . X .
- 1) . XXX . . X .

6 6 4 3 1
→←→←→



미네랄

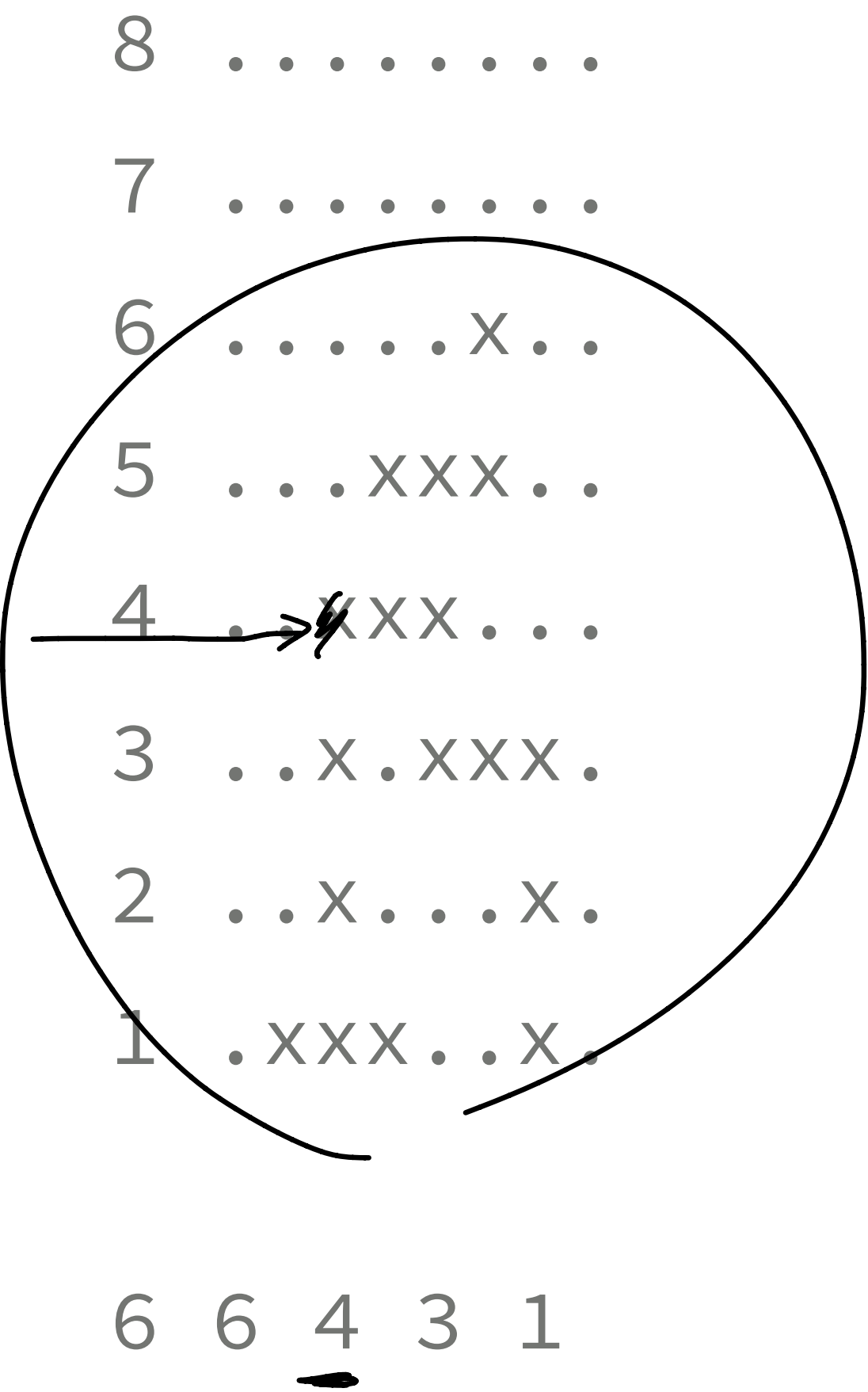
<https://www.acmicpc.net/problem/2933>

8
7
6x ←
5 . . .xxx.
4 . .xxx.
3 . .x.xxx.
2 . .x...x.
1 .xxx..x.

6 6 4 3 1

미네랄

<https://www.acmicpc.net/problem/2933>



미네랄

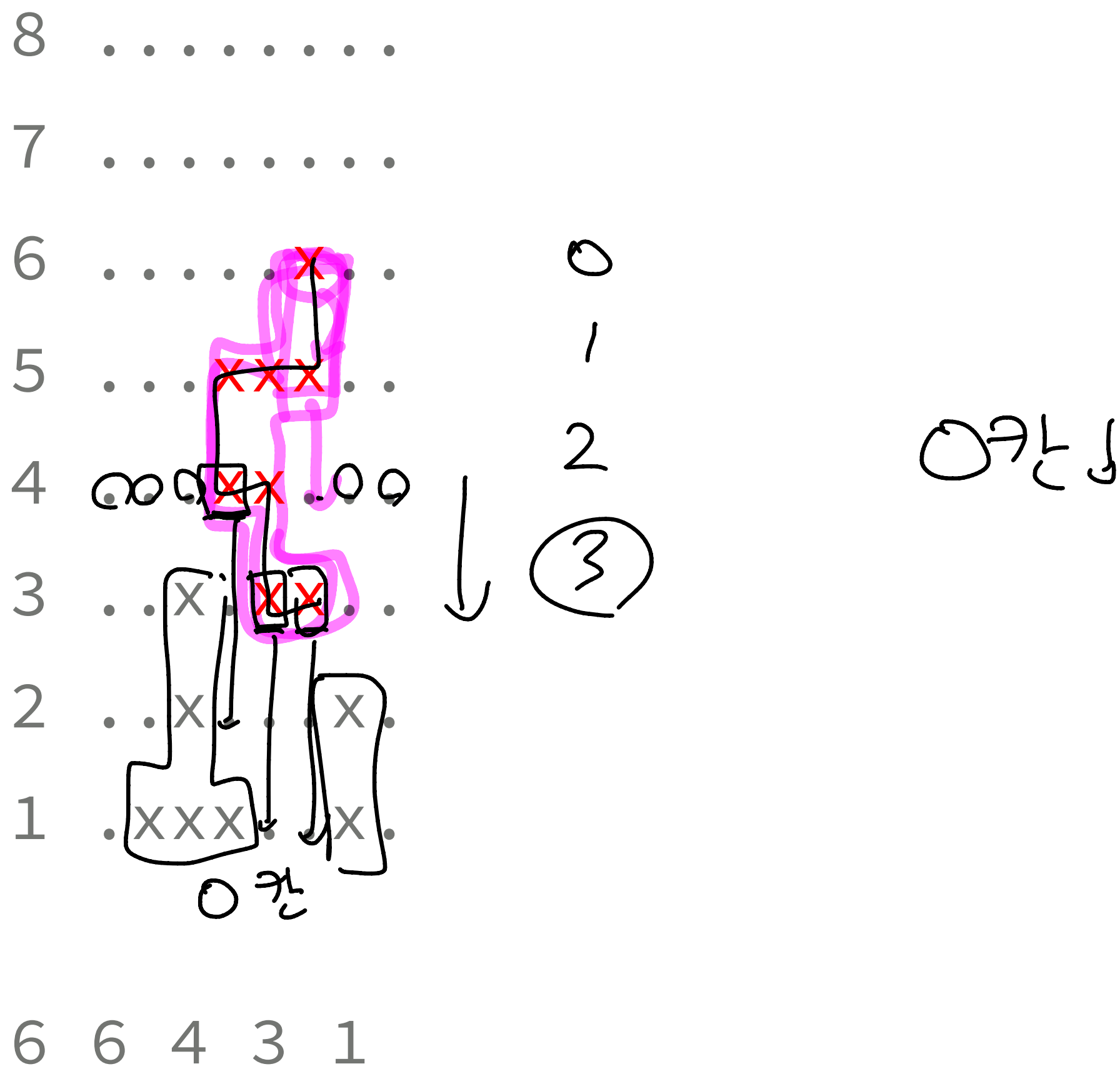
<https://www.acmicpc.net/problem/2933>

8
7
6 X . .
5 . . . XXX . .
4 . . . XX . . .
3 . . X . XX ~~h~~ . ←
2 . . X . . . X .
1 . XXX . . X .

6 6 4 3 1
←

미네랄

<https://www.acmicpc.net/problem/2933>



미네랄

<https://www.acmicpc.net/problem/2933>

8
7
6
5
4 x . . .
3 . . x x x x . .
2 . . x x x . x .
1 ~~→~~ x x x x x x .

6 6 4 3 1

미네랄

<https://www.acmicpc.net/problem/2933>

8
7
6
5
4 X . .
3 . . *4* XXX . .
2 . . XXX . X .
1 . . XXXXX .

6 6 4 3 1

미네랄

<https://www.acmicpc.net/problem/2933>

- 입력으로 주어진 대로 시뮬레이션을 해본다.
- 어려운 부분은 클러스터가 떨어지는 것을 구현하는 부분

미네랄

<https://www.acmicpc.net/problem/2933>

- 동굴의 모양이 주어진다
- 두 사람은 던질 막대의 높이를 결정하고 막대를 던진다.
- 이 때, 미네랄을 만나면 미네랄은 모두 파괴되고, 막대는 그 위치에서 이동을 멈춘다.
- 새로운 클러스터가 공중에 떠있으면 중력에 의해서 아래로 내려온다.

미네랄

<https://www.acmicpc.net/problem/2933>

- 소스: <http://codeplus.codes/6ea34ad7d5c748a0a6a76926cf461cd4>

iSharp

<https://www.acmicpc.net/problem/3568>

- 기본 변수형, 배열(`[]`), 참조(`&`), 포인터(`*`)
- 배열, 참조, 포인터는 순서에 상관없이 혼합해서 사용할 수 있다.
- `int`의 참조의 참조의 배열의 포인터도 올바른 타입 `int&&[]*`
- 여러 개의 변수를 한 줄에 정의할 수 있다.
- 공통된 변수형을 쓰고, 각 변수의 이름과 추가적인 변수형을 쓴다.
- `int& a*[]&, b, c*;`
- `a`의 타입은 `int&&[]*`, `b`는 `int&`, `c`는 `int&*`
- 변수의 오른쪽에 있는 변수형은 순서를 뒤집어서 왼쪽에 붙일 수 있다.
- `int *& a`는 `int a*&`와 같다

iSharp

51

<https://www.acmicpc.net/problem/3568>

- 한 줄에 써있는 변수 선언문을 하나씩 선언하게 바꾸는 문제
- `int& a*[]&, b, c*;`
- 를
- `int&&[]* a;`
- `int& b;`
- `int&* c;`
- 로 바꾸는 문제

iSharp

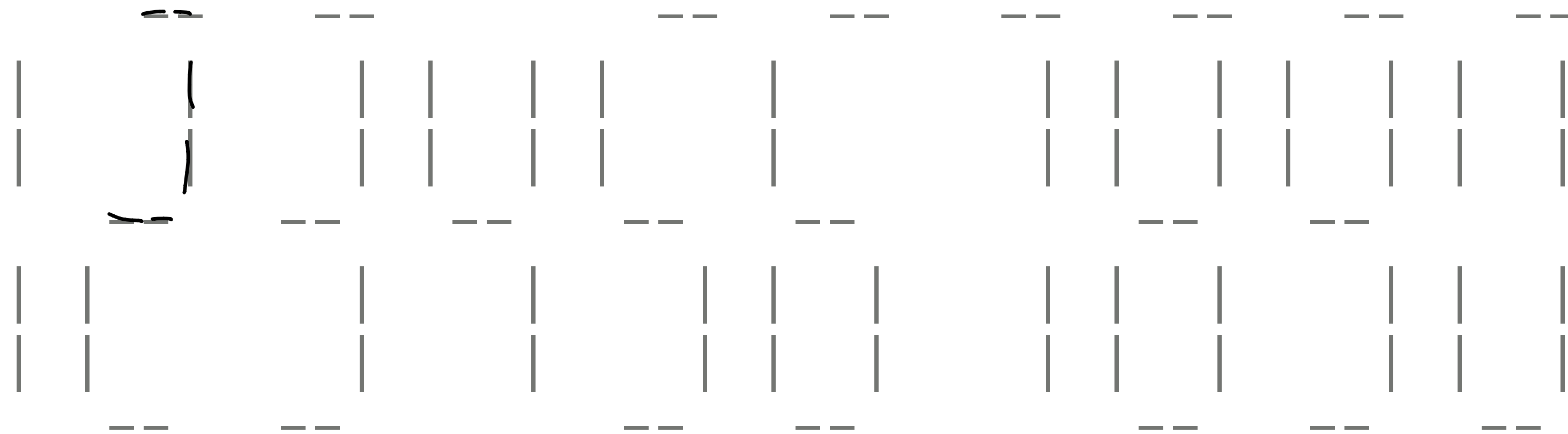
52

<https://www.acmicpc.net/problem/3568>

- 소스: <http://codeplus.codes/7cecc52a12c24dbb982897be559bb147>

- S와 N이 주어졌을 때, 문제에 나와있는 형식으로 출력하는 문제

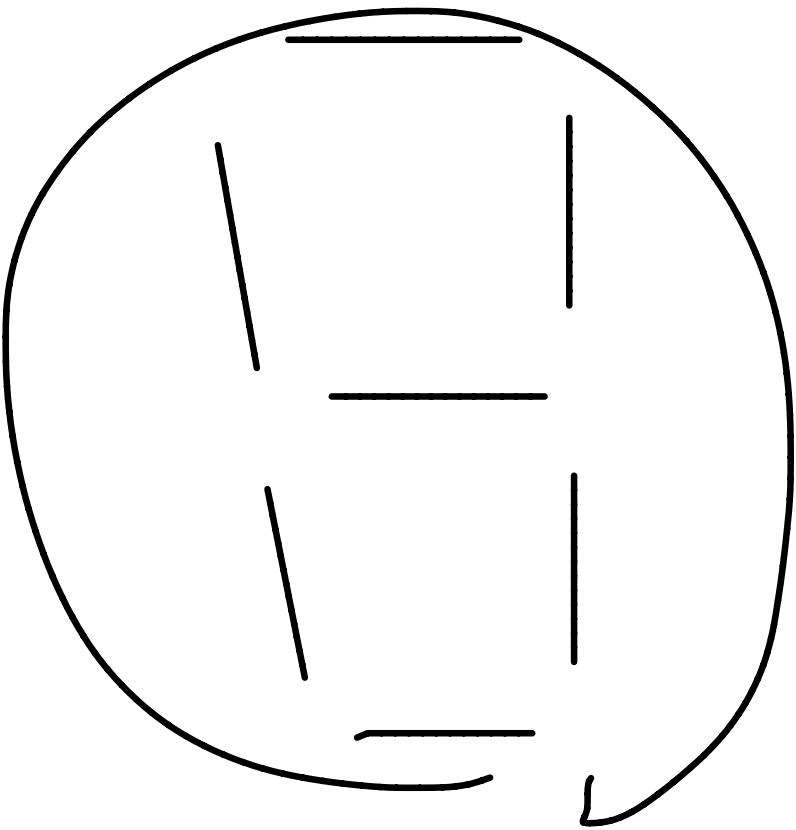
- 2 1234567890



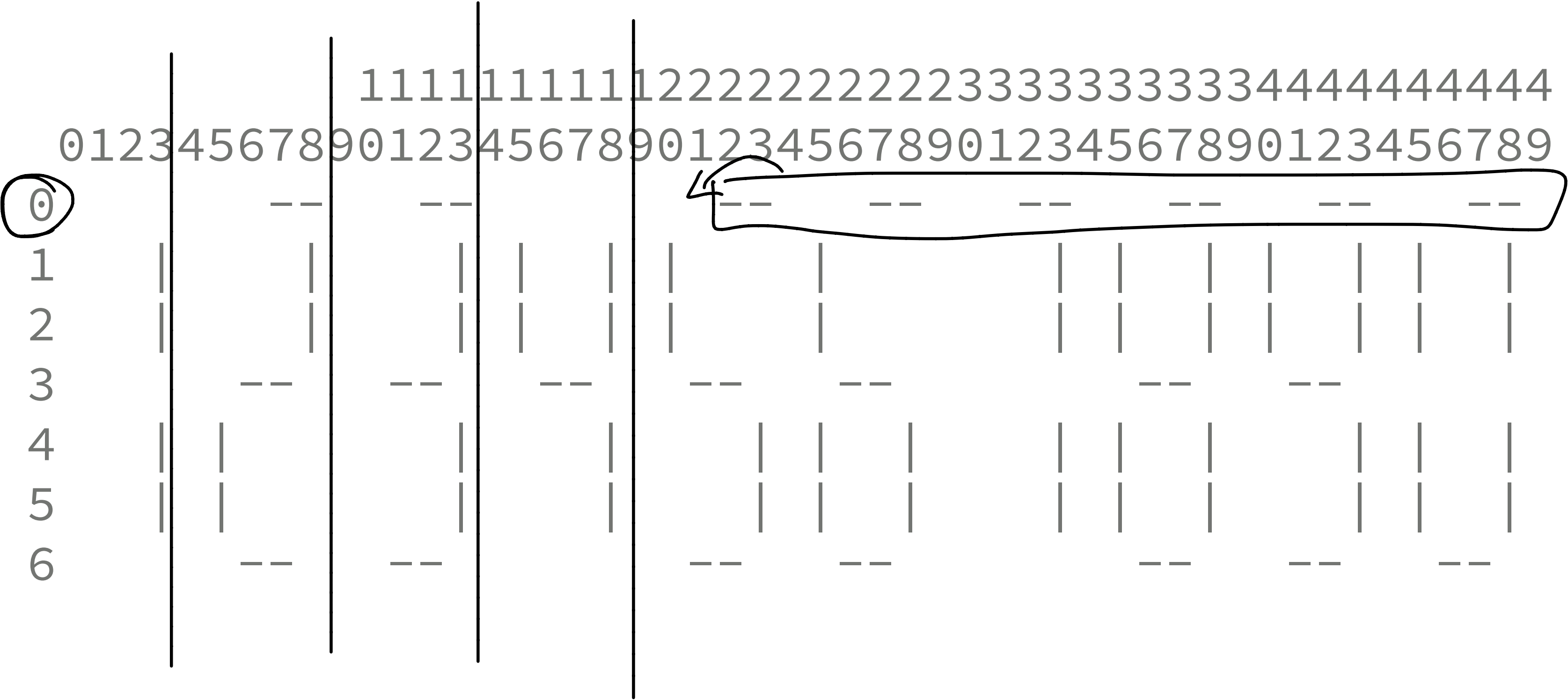
LCD Test

<https://www.acmicpc.net/problem/2290>

- S와 N이 주어졌을 때, 문제에 나와있는 형식으로 출력하는 문제
- 2 1234567890



XS 4m



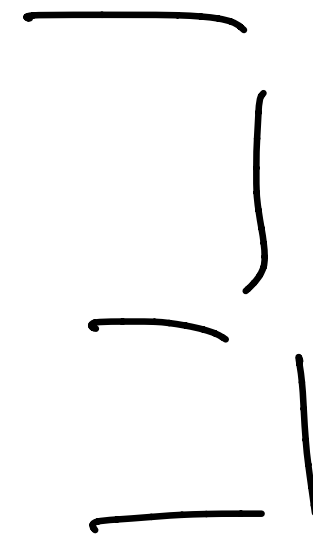
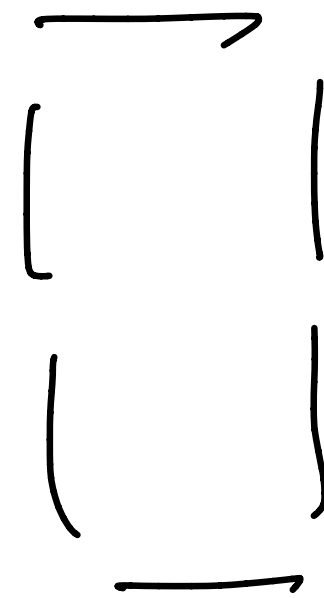
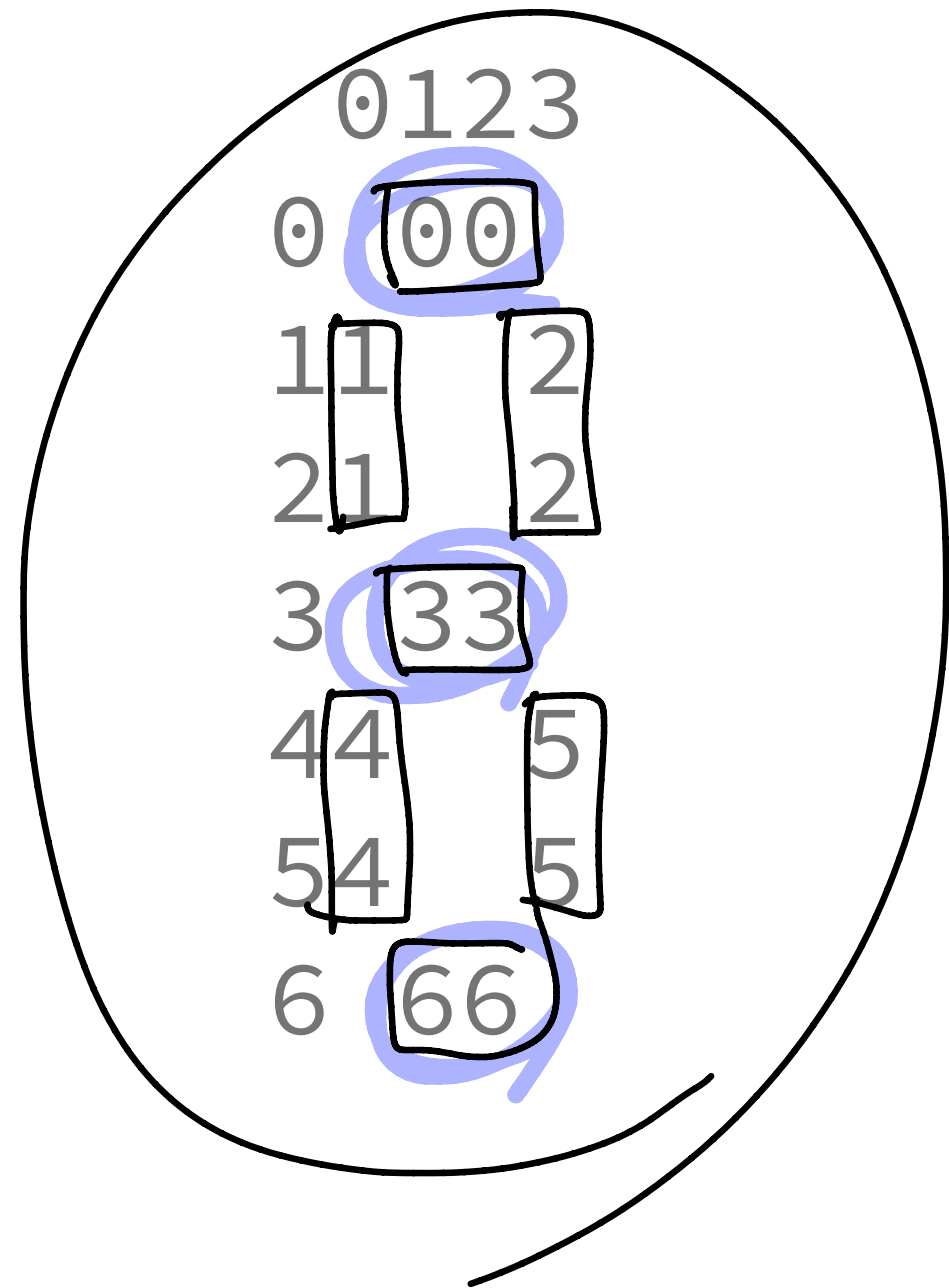
- S와 N이 주어졌을 때, 문제에 나와있는 형식으로 출력하는 문제
- 2 1234567890

[illegible]

LCD Test

<https://www.acmicpc.net/problem/2290>

56



LCD Test

<https://www.acmicpc.net/problem/2290>

- 소스: <http://codeplus.codes/06d03504cc494fd9b5537fb420d66277>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.